Management Guide

NetIQ® AppManager® for Avaya™ Communication Manager

February 2020

NetIQ

Legal Notice

For information about legal notices, trademarks, disclaimers, warranties, export and other use restrictions, U.S. Government rights, patent policy, and FIPS compliance, see https://www.netiq.com/company/legal/.

Copyright (C) 2020 NetIQ Corporation. All rights reserved.

Contents

About this Book and the Library **5**
About NetIQ Corporation **7**

1 Installing and Configuring AppManager for Avaya Communication Manager **9**

System Requirements . . . 9
Scalability Considerations . . . 10
Installing the Module . . . 11
Deploying the Module with Control Center . . . 13
Silently Installing the Module . . . 13
Configuring SNMP Community Strings . . . 14
Configuring Communication Manager To Send RTCP Packets and CDRs . . . 15
Updating the 46xxsettings.txt File . . . 25
Discovering Avaya Communication Manager Resources . . . 26
Configuring Unique Port Numbers for Multiple Communication Managers . . . 32
Upgrading Knowledge Script Jobs . . . 33
Setting Up MSDE . . . 34
Understanding the Log Configuration File . . . 36
Troubleshooting . . . 37

2 Introducing AppManager for Avaya Communication Manager **41**

Features and Benefits . . . 41
Proxy Architecture and Deployment . . . 42
Counting AppManager Licenses . . . 43

3 AvayaCM Knowledge Scripts **45**

AddMIB . . . 46
AddPhone . . . 48
Announcements . . . 50
AttendantCalls . . . 52
CallActivity . . . 55
CallFailures . . . 57
CallQuality . . . 61
CallQuery . . . 67
CPU_Usage . . . 69
DeleteCDRRecords . . . 71
ESS_Status . . . 72
H248GatewayStatus . . . 73
HuntGroupUsage . . . 76
LSP_Status . . . 79
PhoneConnectivity . . . 80
PhoneDeregistrations . . . 82
PhoneInventory . . . 84
PhoneQuality . . . 87
RegisteredResources . . . 91
RemovePhone . . . 96

RetrieveConfigData 96
SecurityViolations 98
SetupSupplementalDB 101
SNMPTrap 103
SystemUptime 108
TrunkGroupUsage 109
Recommended Knowledge Script Group 112

About this Book and the Library

The NetIQ AppManager product (AppManager) is a comprehensive solution for managing, diagnosing, and analyzing performance, availability, and health for a broad spectrum of operating environments, applications, services, and server hardware.

AppManager provides system administrators with a central, easy-to-use console to view critical server and application resources across the enterprise. With AppManager, administrative staff can monitor computer and application resources, check for potential problems, initiate responsive actions, automate routine tasks, and gather performance data for real-time and historical reporting and analysis.

Intended Audience

This guide provides information for individuals responsible for installing an AppManager module and monitoring specific applications with AppManager.

Other Information in the Library

The library provides the following information resources:

Installation Guide for AppManager

Provides complete information about AppManager pre-installation requirements and step-by-step installation procedures for all AppManager components.

User Guide for AppManager Control Center

Provides complete information about managing groups of computers, including running jobs, responding to events, creating reports, and working with Control Center. A separate guide is available for the AppManager Operator Console.

Administrator Guide for AppManager

Provides information about maintaining an AppManager management site, managing security, using scripts to handle AppManager tasks, and leveraging advanced configuration options.

Upgrade and Migration Guide for AppManager

Provides complete information about how to upgrade from a previous version of AppManager.

Management guides

Provide information about installing and monitoring specific applications with AppManager.

Help

Provides context-sensitive information and step-by-step guidance for common tasks, as well as definitions for each field on each window.

The AppManager library is available in Adobe Acrobat (PDF) format from the AppManager Documentation page of the NetIQ Web site.

About NetIQ Corporation

We are a global, enterprise software company, with a focus on the three persistent challenges in your environment: Change, complexity and risk—and how we can help you control them.

Our Viewpoint

Adapting to change and managing complexity and risk are nothing new

In fact, of all the challenges you face, these are perhaps the most prominent variables that deny you the control you need to securely measure, monitor, and manage your physical, virtual, and cloud computing environments.

Enabling critical business services, better and faster

We believe that providing as much control as possible to IT organizations is the only way to enable timelier and cost effective delivery of services. Persistent pressures like change and complexity will only continue to increase as organizations continue to change and the technologies needed to manage them become inherently more complex.

Our Philosophy

Selling intelligent solutions, not just software

In order to provide reliable control, we first make sure we understand the real-world scenarios in which IT organizations like yours operate — day in and day out. That's the only way we can develop practical, intelligent IT solutions that successfully yield proven, measurable results. And that's so much more rewarding than simply selling software.

Driving your success is our passion

We place your success at the heart of how we do business. From product inception to deployment, we understand that you need IT solutions that work well and integrate seamlessly with your existing investments; you need ongoing support and training post-deployment; and you need someone that is truly easy to work with — for a change. Ultimately, when you succeed, we all succeed.

Our Solutions

- Identity & Access Governance
- Access Management
- Security Management
- Systems & Application Management
- Workload Management
- Service Management

Contacting Sales Support

For questions about products, pricing, and capabilities, contact your local partner. If you cannot contact your partner, contact our Sales Support team.

Worldwide:	www.netiq.com/about_netiq/officelocations.asp
United States and Canada:	1-888-323-6768
Email:	info@netiq.com
Web Site:	www.netiq.com

Contacting Technical Support

For specific product issues, contact our Technical Support team.

Worldwide:	www.netiq.com/support/contactinfo.asp
North and South America:	1-713-418-5555
Europe, Middle East, and Africa:	+353 (0) 91-782 677
Email:	support@netiq.com
Web Site:	www.netiq.com/support

Contacting Documentation Support

Our goal is to provide documentation that meets your needs. The documentation for this product is available on the NetIQ Web site in HTML and PDF formats on a page that does not require you to log in. If you have suggestions for documentation improvements, click **comment on this topic** at the bottom of any page in the HTML version of the documentation posted at www.netiq.com/documentation. You can also email Documentation-Feedback@netiq.com. We value your input and look forward to hearing from you.

Contacting the Online User Community

NetIQ Communities, the NetIQ online community, is a collaborative network connecting you to your peers and NetIQ experts. By providing more immediate information, useful links to helpful resources, and access to NetIQ experts, NetIQ Communities helps ensure you are mastering the knowledge you need to realize the full potential of IT investments upon which you rely. For more information, visit http://community.netiq.com.

1 Installing and Configuring AppManager for Avaya Communication Manager

This chapter provides installation instructions, and describes system requirements and configuration information for AppManager for Avaya Communication Manager.

This chapter assumes you have AppManager installed. For more information about installing AppManager or about AppManager system requirements, see the *Installation Guide for AppManager*, which is available on the AppManager Documentation page.

System Requirements

For the latest information about supported software versions and the availability of module updates, visit the AppManager Supported Products page. Unless noted otherwise, this module supports all updates, hotfixes, and service packs for the releases listed below.

AppManager for Avaya Communication Manager has the following system requirements:

Software/Hardware	Version
NetIQ AppManager installed on the AppManager repository (QDB) computer, on each proxy agent computer, and on all console computers	7.0 or later Support for Windows Server 2008 on AppManager 7.x requires AppManager Windows Agent hotfix 71704 or later. For more information, see the AppManager Suite Hotfixes page.
Avaya Aura Communication Manager (also known as Avaya Communication Manager)	8.0, 7.1.3, 7.0, 6.3, 6.2, 6.1, 6.0, 5.x, or 4.x **NOTE:** For version 5.2.1, install Avaya Communication Manager Service Pack 3 or later to improve SNMP performance. For more information, see the Avaya Aura Communication Manager 5.2.1 SP3 Release Notes.
Microsoft Windows operating system installed on the agent computers	One of the following versions: ◆ Windows Server 2019 ◆ Windows Server 2016 ◆ Windows Server 2012 R2 ◆ Windows Server 2012 ◆ Windows 8 (32-bit and 64-bit) ◆ Windows Server 2008 R2 ◆ Windows Server 2008 (32-bit and 64-bit) ◆ Windows 7 (32-bit and 64-bit)

Software/Hardware	Version
Microsoft SQL Server installed on the proxy agent computers	One of the following versions, to enable the functionality of the Avaya CM supplemental database: ◆ SQL Server Express edition ◆ SQL Server 2017 ◆ SQL Server 2016 ◆ SQL server 2014 ◆ SQL server 2012 ◆ SQL Server 2008 R2
AppManager for Microsoft Windows module installed on repository, proxy agent, and console computers	7.6.170.0 or later. For more information, see the AppManager Module Upgrades & Trials page.
Microsoft .NET Framework	4.0 or above.
Microsoft SQL Server Native Client 11.0 **(for TLS 1.2 support)**	11.3.6538.0 or later **NOTE:** The SQL Server Native client can be installed from this Microsoft download link.

NOTE: If you want TLS 1.2 support and are running AppManager 9.1 or 9.2, then you are required to perform some additional steps. To know about the steps, see the article.

Scalability Considerations

Consider the following before installing the AppManager for Avaya Communication Manager module:

- For CDR-processing tasks for the AvayaCM_Call* Knowledge Scripts, the primary scalability factor is the number of *active* phones you are monitoring. Active phones are configured phones that are off the hook and in use. Active phones generate call detail records (CDRs), which increase CPU usage on the proxy agent computer. Inactive phones, those that are configured but not in use, do not increase CPU usage on the proxy agent computer and therefore are not a scalability consideration.

  One proxy agent computer can support the monitoring of approximately 1000 active phones in one cluster or across multiple clusters. It is possible to monitor more active phones by increasing the interval in which the Communication Manager server sends RTCP packets to the managed object on the proxy computer. Increasing the interval can reduce the strain on the proxy agent computer and allow it to monitor more active phones.
- For SNMP-related tasks for the remaining AvayaCM Knowledge Scripts, one proxy agent computer can support up to five Communication Manager clusters. The efficiency of the SNMP code in these Knowledge Scripts changes only slightly based on the number of phones you are monitoring. From an SNMP perspective, there is little difference in monitoring ten phones or 10,000 phones.
- For the PhoneQuality Knowledge Script, the primary scalability factor is the number of data points the script generates. The script generates seven data points per monitored active phone every 30 seconds, which is a large amount of data when multiplied by thousands of monitored phones. Although the proxy agent computer can handle that much data, the AppManager repository (`QDB`) and management server may eventually be unable to handle the volume of data.

You can decrease the flow of data by turning off some data streams and by changing the value of the *Data collection interval for voice quality metrics* parameter to 1 minute.

The management server can handle a larger volume of data if you increase the size of the map queue. The default size is 5 MB, but you can increase it to 25 MB with no adverse affect on performance. An even larger value may work in your environment.

To change the size of the map queue:

1. On the management server computer, open the Registry Editor and navigate to `HKEY_LOCAL_MACHINE\SOFTWARE\NetIQ\AppManager\4.0\NetIQms\Config`.
2. In the right pane, double-click **PIOC Data Map File Size MB**.
3. In the Edit DWORD Value dialog box, change the value in the **Value data** field.
4. Restart the NetIQ AppManager Management Service, `NetIQms.exe`. The change in file size does not take effect until you restart the service.

- Based on the preceding scalability considerations, your proxy agent computer should meet the following minimum hardware requirements:
  - Dual Pentium 4 processor
  - 3.4 GHz
  - 2 GB RAM

Installing the Module

Run the module installer on the proxy agent computers you want to monitor to install the agent components, and run the module installer on all console computers to install the Help and console extensions.

NOTE: Installing the module automatically installs NetIQ SNMP Trap Receiver (Trap Receiver). For more information, see "Working with NetIQ SNMP Trap Receiver" on page 105.

Access the `AM70-AvayaCM-8.x.x.0.msi` module installer from the `AM70_AvayaCM_8.x.x.0.` self-extracting installation package on the AppManager Module Upgrades & Trials page.

You can install the Knowledge Scripts into local or remote AppManager repositories (QDBs). Install these components only once per QDB. The module installer now installs Knowledge Scripts for each module directly into the QDB instead of to the `\AppManager\qdb\kp` folder as in previous releases of AppManager.

You can install the module manually, or you can use Control Center to deploy the module on a remote computer where an agent is installed. For more information, see "Deploying the Module with Control Center" on page 13. However, if you do use Control Center to deploy the module, Control Center only installs the *agent* components of the module. The module installer installs the QDB and console components as well as the agent components on the agent computer.

To install the module manually:

1. On all proxy agent computers, stop the NetIQ AppManager Client Resource Monitor (`NetIQmc`) service to ensure that any existing version of `qNQAvayaCM.dll` is updated correctly during installation of the module.
2. Double-click the module installer `.msi` file.
3. Accept the license agreement.

4. Review the results of the pre-installation check. You can expect one of the following three scenarios:
   - *No AppManager agent is present*. In this scenario, the pre-installation check fails, and the installer does not install agent components.
   - *An AppManager agent is present, but some other prerequisite fails*. In this scenario, the default is to not install agent components because of one or more missing prerequisites. However, you can override the default by selecting **Install agent component locally**. A missing application server for this particular module often causes this scenario. For example, installing the AppManager for Microsoft SharePoint module requires the presence of a Microsoft SharePoint server on the selected computer.
   - *All prerequisites are met*. In this scenario, the installer will install the agent components.
5. To install the Knowledge Scripts into the QDB:
   - **5a** Select **Install Knowledge Scripts** to install the repository components, including the Knowledge Scripts, object types, and SQL stored procedures.
   - **5b** Specify the SQL Server name of the server hosting the QDB, as well as the case-sensitive QDB name.
6. (Conditional) If you use Control Center 7.x, run the module installer for each QDB attached to Control Center.
7. (Conditional) If you use Control Center 8.x or later, run the module installer only for the primary QDB, and Control Center automatically replicates this module to secondary QDBs.
8. Run the module installer on all console computers to install the Help and console extensions.
9. Run the module installer on all proxy agent computers to install the agent components.
10. Configure all necessary SNMP community strings in AppManager Security Manager to enable access of remote Communication Managers and to enable the functionality of Trap Receiver. For more information, see "Configuring SNMP Community Strings" on page 14.
11. Enable Communication Manager to send RTCP packets. For more information, see "Configuring Communication Manager To Send RTCP Packets and CDRs" on page 15.
12. Update configuration settings for your SIP phones to identify the proxy agent computer. For more information, see "Updating the 46xxsettings.txt File" on page 25.
13. (Conditional) If you have not discovered Avaya Communication Manager resources, run the Discovery_AvayaCM Knowledge Script on all proxy agent computers where you installed the module. For more information, see "Discovering Avaya Communication Manager Resources" on page 26.
14. To get the updates provided in this release, upgrade any running Knowledge Script jobs. For more information, see "Upgrading Knowledge Script Jobs" on page 33.
15. Run the SetupSupplementalDB Knowledge Script to:
    - **15a** Apply improved SQL indices to the CDR and Traceroute tables in the Avaya CM supplemental database.
    - **15b** Update the CDR table in the Avaya CM supplemental database to allow the database to store Facilities Restriction Level (FRL) codes.

After the installation has completed, you can find a record of problems encountered in the `AvayaCM_Install.log` file, located in the `\NetIQ\Temp\NetIQ_Debug\<`*ServerName*`>` folder.

IMPORTANT: Configure the gateways to send traps directly to the trap receiver instead of forwarding them through the Communications Manager. For additional information about how to configure the gateways, see https://downloads.avaya.com/css/P8/documents/100059440.

Deploying the Module with Control Center

You can use Control Center to deploy the module on a remote computer where an agent is installed. This topic briefly describes the steps involved in deploying a module and provides instructions for checking in the module installation package. For more information, see the *Control Center User Guide for AppManager*, which is available on the AppManager Documentation page.

Deployment Overview

This section describes the tasks required to deploy the module on an agent computer.

To deploy the module on a proxy agent computer:

1. Verify the default deployment credentials.
2. Check in an installation package. For more information, see "Checking In the Installation Package" on page 13.
3. Configure an email address to receive notification of a deployment.
4. Create a deployment rule or modify an out-of-the-box deployment rule.
5. Approve the deployment task.
6. View the results.

Checking In the Installation Package

You must check in the installation package, `AM70-AvayaCM-8.x.x.0.xml`, before you can deploy the module on an agent computer.

To check in a module installation package:

1. Log on to Control Center using an account that is a member of a user group with deployment permissions.
2. Navigate to the **Deployment** tab (for AppManager 8.x or later) or **Administration** tab (for AppManager 7.x).
3. In the Deployment folder, select **Packages**.
4. On the Tasks pane, click **Check in Deployment Packages** (for AppManager 8.x or later) or **Check in Packages** (for AppManager 7.x).
5. Navigate to the folder where you saved `AM70-AvayaCM-8.x.x.0.xml` and select the file.
6. Click **Open**. The Deployment Package Check in Status dialog box displays the status of the package check in.

Silently Installing the Module

To silently (without user intervention) install a module using the default settings, run the following command from the folder in which you saved the module installer:

```
msiexec.exe /i "AM70-AvayaCM-8.x.x.0.msi" /qn
```

where *x.x* is the actual version number of the module installer.

To get the updates provided in this release, upgrade any running Knowledge Script jobs. For more information, see "Upgrading Knowledge Script Jobs" on page 33.

To create a log file that describes the operations of the module installer, add the following flag to the command noted above:

```
/L* "AM70-AvayaCM-8.x.x.0.msi.log"
```

The log file is created in the directory in which you saved the module installer.

NOTE: To perform a silent install on an AppManager agent running Windows 2008 R2, open a command prompt at the administrative level and select **Run as administrator** before you run the silent install command listed above.

To silently install the module on a remote AppManager repository, you can use Windows authentication or SQL authentication.

Windows authentication:

```
AM70-AvayaCM-8.x.x.0.msi /qn MO_B_QDBINSTALL=1 MO_B_MOINSTALL=0
MO_B_SQLSVR_WINAUTH=1 MO_SQLSVR_NAME=SQLServerName MO_QDBNAME=AM-RepositoryName
```

SQL authentication:

```
AM70-AvayaCM-8.x.x.0.msi /qn MO_B_QDBINSTALL=1 MO_B_MOINSTALL=0
MO_B_SQLSVR_WINAUTH=0 MO_SQLSVR_USER=SQLLogin MO_SQLSVR_PWD=SQLLoginPassword
MO_SQLSVR_NAME=SQLServerName MO_QDBNAME=AM-RepositoryName
```

Configuring SNMP Community Strings

AppManager uses SNMP queries to access remote Communication Manager servers and to enable the functionality of NetIQ SNMP Trap Receiver. Before discovering Communication Manager resources, enter SNMP community string information into AppManager Security Manager.

AppManager for Avaya Communication Manager supports SNMP versions 1 and 2. Configure Avaya Communication Manager 6.3.100 and 7.x with read-write SNMP community strings to give it access to all SNMP objects. If you configure it with read-only access, you will receive a warning and only a partial discovery will occur.

Configuring Community Strings for Remote Communication Managers

Configure SNMP community string information for each Communication Manager you want to monitor.

On the Custom tab in Security Manager, complete the following fields:

Field	Description
Label	`SNMP`
Sub-label	Indicate whether the community string information will be used for a single device or for all devices: ◆ *For a single Communication Manager*, type <*device IP address* or *hostname*>. The address or hostname must match the address or hostname you provide in the parameters for the Discovery_AvayaCM Knowledge Script. ◆ *For all Communication Managers*, type `default`.
Value 1	The appropriate read-write community string value, such as `private` or `public`. **NOTE:** Avaya Call Manager Clusters using the AVAYA-AURA-CM-MIB require read/write community string permission. All other Avaya clusters require read-only permissions.

Configuring Community Strings for Trap Receiver Functionality

Configure SNMP community string information for each Trap Receiver device that will send traps to the proxy agent computer.

On the Custom tab in Security Manager, complete the following fields:

Field	Description
Label	`SNMPTrap`
Sub-label	Provide the IP address or hostname of the device on which Trap Receiver is installed. **NOTE:** If you have already run the Discovery_AvayaCM Knowledge Script, use the same IP address or hostname that is displayed for the AvayaCM object in the TreeView.
Value 1	Provide the community string name included in each trap sent by Trap Receiver, such as `private` or `public`.

Configuring Communication Manager To Send RTCP Packets and CDRs

Several Knowledge Scripts need Communication Manager to send RTCP (Real-time Transport Control Protocol) packets and call detail records (CDRs) to the AppManager agent on the proxy computer:

- CallActivity, which needs both RTCP packets and CDRs
- CallFailures, which needs CDRs

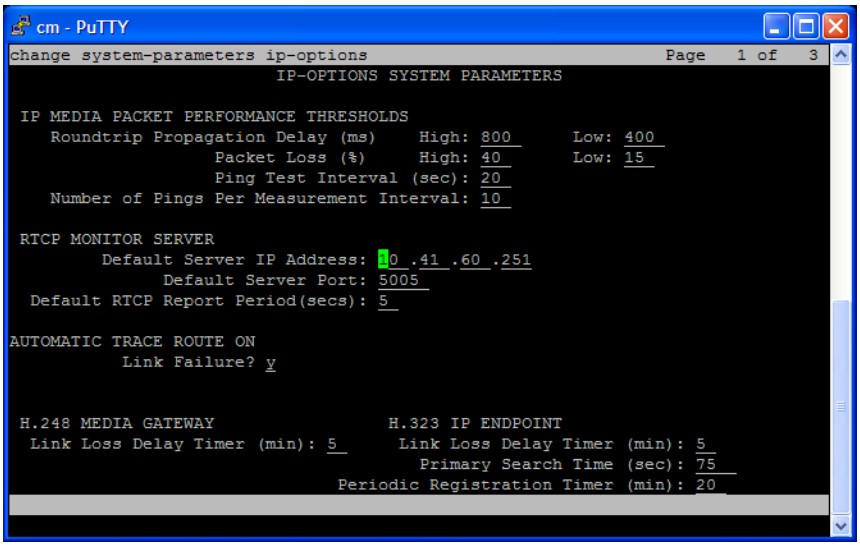
- CallQuality, which needs RTCP packets
- CallQuery, which needs CDRs
- PhoneQuality, which needs RTCP packets

If RTCP packets and CDRs are not sent to the AppManager agent, these scripts cannot collect data or raise threshold-crossing events.

Use the Communication Manager System Administrator Terminal (SAT) to configure RTCP packets and CDRs. Commands for this screen-based terminal application generally start with a verb such as “display,” “list,” and “change.” In the following topics, use “change” commands to implement the RTCP and CDR configuration. The procedures in these topics assume you know how to navigate in SAT.

Step 1: Configuring IP Address and Port for RTCP Packets

Use the IP-Options System Parameters screen to configure the IP address and port to which Avaya IP phones will send RTCP packets. To access this screen, execute the following command:
`change system-parameters ip-options`



Complete the fields on the screen according to the information in the following table. Press [Esc + E] to save your changes and return to the command line.

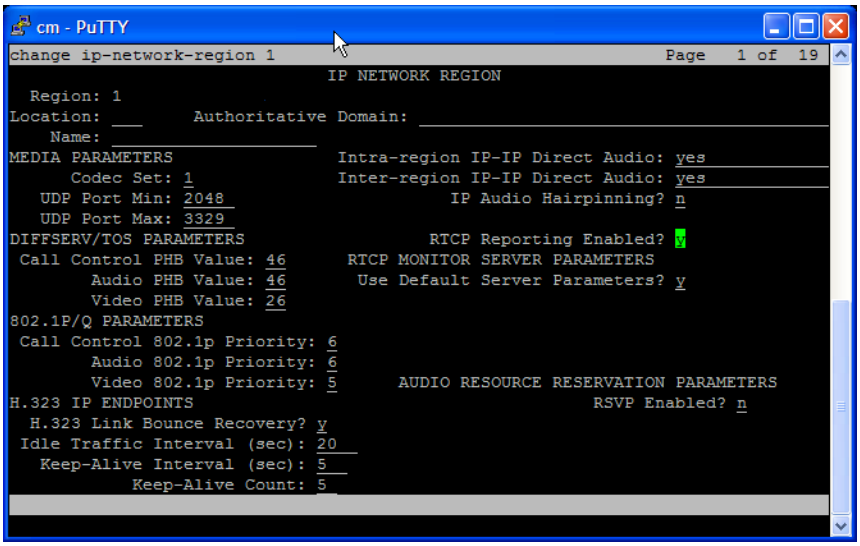
Field	Description
Default Server IP Address	Type the IP address of the computer acting as proxy for Communication Manager servers. Avaya IP phones must be able to send RTCP packets to this computer. Any firewall between the phones and the proxy computer must be configured to allow UDP packets to flow through it in the direction from phones to proxy computer.

Field	Description
Default Server Port	Type the UDP port number on the proxy computer to which Avaya IP phones send RTCP packets. The default port number is 5005. You can change it to any other valid port number, but remember that RTCP specifications require an odd, not even, port number. **NOTE:** For each Communication Manager being monitored by the proxy computer, you must specify a different UDP port number here. AppManager uses the port number to determine which RTCP packets belong to a given Communication Manager. For more information, see "Configuring Unique Port Numbers for Multiple Communication Managers" on page 32.
Default RTCP Report Period	Use this field to specify the frequency with which Avaya IP phones send RTCP packets to the proxy computer. The default is every 30 seconds. Valid values are between 5 and 30. **Important concepts** ◆ This value must match the value you specify in the **Value 3** field in "Configuring Unique Port Numbers for Multiple Communication Managers" on page 32. For information about what happens when the values do not match, see "PhoneQuality Script Not Collecting Complete Data" on page 37. ◆ The PhoneQuality script produces the best results when this value is low. A 5-second interval is recommended. However, if you are monitoring many phones, the AppManager agent may become overloaded if many phones send packets at 5-second intervals. In this case, you can specify a less-frequent interval.

Step 2: Changing or Disabling RTCP Per IP Network Region

All Avaya IP phones belong to a particular IP network region, which is a set of IP address ranges. By default, the phones associated with a region send RTCP packets to the server specified in "Step 1: Configuring IP Address and Port for RTCP Packets" on page 16. However, you can specify that phones associated with a particular IP network region send their RTCP packets to a different server, with even a different frequency, or you can disable the sending mechanism altogether.

Use the IP Network Region screen to change a region's RTCP configuration. To access this screen, execute the following command: `change ip-network-region` *region number*]



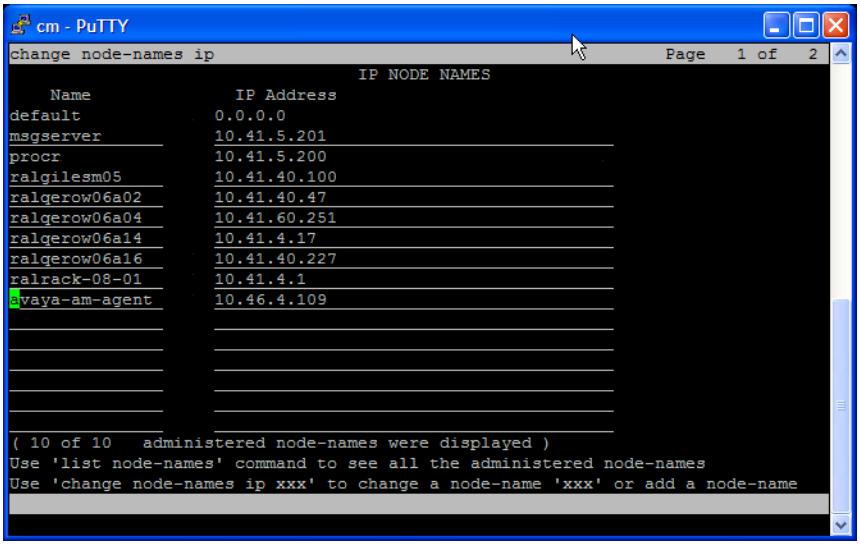
Complete the fields on the screen according to the information in the following table. Press [`Esc` + `E`] to save your changes and return to the command line.

Field	Description
RTCP Reporting Enabled?	Set to **y** to enable RTCP reporting. Set to **n** to disable the RTCP reporting mechanism altogether.
Use Default Server Parameters?	Set to **y** to use the values specified in "Step 1: Configuring IP Address and Port for RTCP Packets" on page 16. Set to **n** to use different values. If you set to **n**, press [`Esc` + `E`] to save your changes. Then execute the `change ip-network-region` [*region number*] command again. New fields are displayed on the screen. ◆ **Server IP Address** - Type the IP address of the computer to which you want to send RTCP packets. ◆ **Server Port** - Type the UDP port number on the computer to which you want to send RTCP packets. ◆ **RTCP Report Period** - Specify the frequency with which Avaya phones will send RTCP packets to the computer. The frequency you enter here overrides any global frequency you set in "Step 1: Configuring IP Address and Port for RTCP Packets" on page 16. The recommended frequency is 5 seconds.

Step 3: Assigning a Node Name to the Proxy Agent Computer

Assign a node name to the proxy agent computer's IP address. The node name enables Communication Manager to recognize the proxy agent computer and send CDRs to it.

Use the IP Node Names screen to assign a node name to the proxy agent computer's IP address. To access this screen, execute the following command: `change node-names ip`



Complete the fields on the screen according to the information in the following table. Press [`Esc` + `E`] to save your changes and return to the command line.

Field	Description
Name	Navigate to an empty **Name** field and type the host name of the proxy computer.
IP Address	In the associated **IP Address** field, type the IP address of the proxy computer.

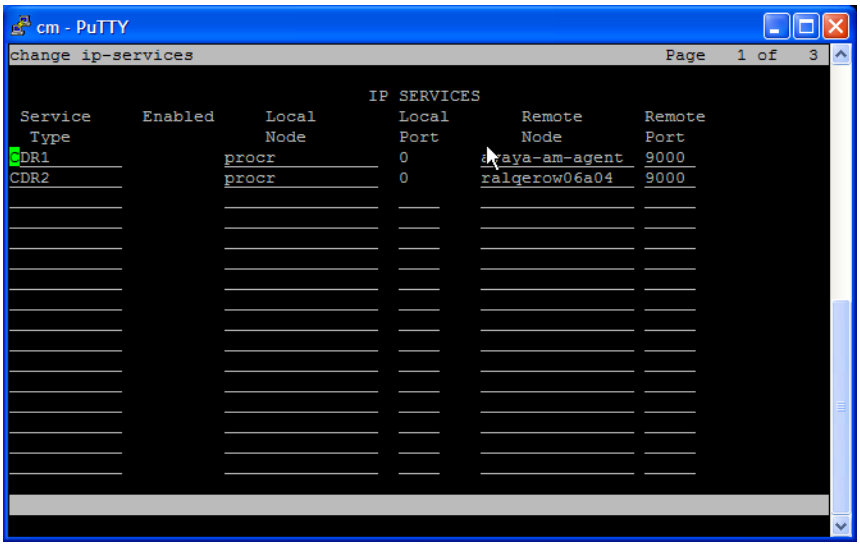
Step 4: Defining the CDR Link to the Proxy Agent Computer

Communication Manager can send CDRs to one or two endpoint computers (or recipients; not to be confused with NetIQ Performance Endpoints): CDR1 and CDR2. You need to define the link between these endpoint computers and the proxy agent computer.

NOTE

- One of the endpoint computers must be the proxy agent computer.
- You configure the format of the endpoint computers in "Step 5: Configuring CDR Endpoints and Format" on page 20.

Use the IP Services screen to associate CDR1 or CDR2. To access this screen, execute the following command: `change ip-services`



Complete the fields on the screen according to the information in the following table. Press [`Esc + E`] to save your changes and return to the command line.

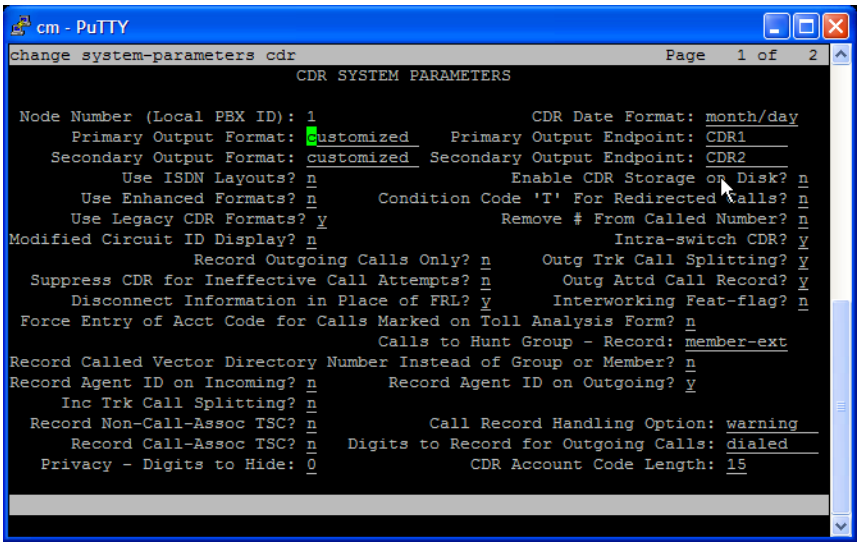
Field	Description
Service Type	Type `CDR1` or `CDR2`, as appropriate.
Local Node	Type `procr`.
Local Port	Type `0`.
Remote Node	Type the node name you assigned in "Step 3: Assigning a Node Name to the Proxy Agent Computer" on page 19.
Remote Port	Type the TCP port number to which CDRs will be sent on the proxy agent computer. **Notes** ◆ For each Communication Manager being monitored by the proxy agent computer, specify a different port number in this field. For more information, see "Configuring Unique Port Numbers for Multiple Communication Managers" on page 32. ◆ Any firewall between Communication Manager and the proxy agent computer must allow TCP connections from Communication Manager to the proxy agent computer on this port number.

Step 5: Configuring CDR Endpoints and Format

The CDR format can be different for each endpoint computer. Communication Manager offers several predefined formats and allows for the creation of a custom format. The CDRs sent to the AppManager agent on the proxy computer need a custom format.

NOTE: If CDR1 is formatted for a billing system, the formatting procedure is different. For more information, see "Step 7 (Optional): Configuring CDR Format if CDR1 is Formatted For a Billing System" on page 23.

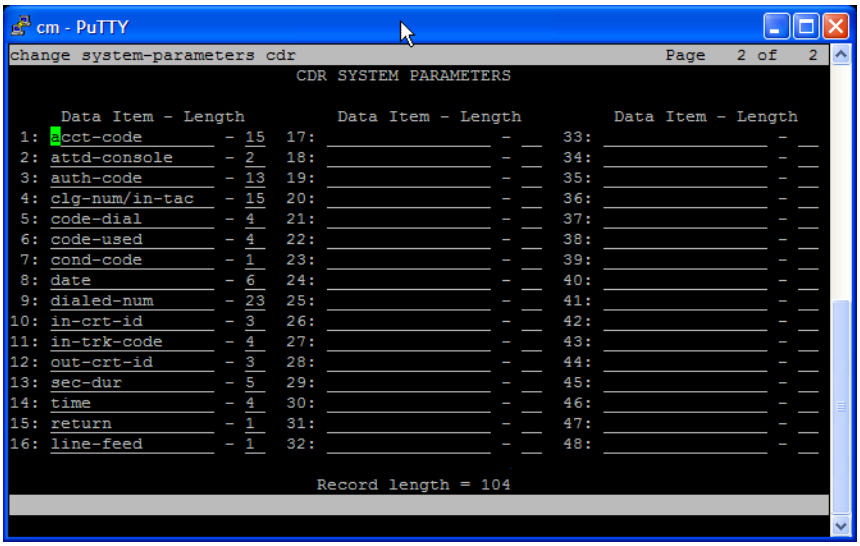
Use the CDR System Parameters screen to assign an endpoint and to configure CDR format. To access this screen, execute the following command: `change system-parameters cdr`



Use the **Primary Output** fields to configure one set of endpoint information and the **Secondary Output** fields to configure the other. The following table provides directions for configuring Primary Output; the process is the same for configuring Secondary Output.

Field	Description
Primary Output Format	Type `customized`.
Primary Output Endpoint	Type `CDR1` or `CDR2`. Use the label that is *not* being used by the **Secondary Output Endpoint** field.
Intra-switch CDR?	Set to **y** to ensure that a CDR contains all calls between two phones. If you set this field to **n**, a CDR will contain data about incoming and outgoing calls, but no data about internal calls. If you set this field to **y**, also specify the phone extensions for which intra-switch CDRs will be created. For more information, see "Step 8 (Optional): Specifying Intra-Switch Extension Numbers" on page 25.
Suppress CDR for Ineffective Call Attempts?	Set to **n** to ensure CDRs contain information about failed calls. If you set this field to **y**, CDRs contain only data about successful calls and the CallFailures script will have no data to retrieve.

Press [Esc + N] to navigate to the next page of the screen.



Use the fields on this page to customize the CDR format. In rows 1 through 16, type the Data Item and Length values as indicated in the following table. Press [Esc + E] to save your changes and return to the command line.

Row	Data Item	Length
1	acct-code	15
2	attd-console	2
3	auth-code	13
4	clg-num/in-tac	15
5	code-dial	4
6	code-used	4
7	cond-code	1
8	date	6
9	dialed-num	23
10	in-crt-id	3
11	in-trk-code	4
12	out-crt-id	3
13	sec-dur	5
14	time	4
15	return	1
16	line-feed	1

Step 6 (Optional): Configuring CDR Parameters to Filter by FRL Codes

If you want to filter by some or all of the Facilities Restriction Level (FRL) codes, you can provide a comma-separated list of the codes you want to filter by for the *Include only these FRL codes* parameter of the CallFailures Knowledge Script.

To avoid an error message, run the SetupSupplementalDB Knowledge Script once before running CallFailures. Also, set up FRL at the Avaya System Access Terminal (SAT) interface before running this script, and do not use FRL at the Avaya SAT interface with previous versions of this module.

To set up FRL filtering:

1. To access the CDR System Parameters screen, execute the following command:
   `change system-parameters cdr`
2. Press [`Esc + N`] to navigate to the next page of the CDR System Parameters screen.

```
display system-parameters cdr                                    Page   2 of   2
                            CDR SYSTEM PARAMETERS

     Data Item - Length          Data Item - Length          Data Item - Length
 1: acct-code        - 15  17: line-feed        - 1   33:                  -
 2: attd-console     - 2   18:                  -     34:                  -
 3: auth-code        - 13  19:                  -     35:                  -
 4: clg-num/in-tac   - 15  20:                  -     36:                  -
 5: code-dial        - 4   21:                  -     37:                  -
 6: code-used        - 4   22:                  -     38:                  -
 7: cond-code        - 1   23:                  -     39:                  -
 8: date             - 6   24:                  -     40:                  -
 9: dialed-num       - 23  25:                  -     41:                  -
10: in-crt-id        - 3   26:                  -     42:                  -
11: in-trk-code      - 4   27:                  -     43:                  -
12: out-crt-id       - 3   28:                  -     44:                  -
13: sec-dur          - 5   29:                  -     45:                  -
14: time             - 4   30:                  -     46:                  -
15: frl              - 1   31:                  -     47:                  -
16: return           - 1   32:                  -     48:                  -

                              Record length = 105
```

3. Complete rows 1-14 with the same values as those in step 5.
4. In row 15, specify `frl` as the data item and `1` as the length.
5. In row 16, specify `return` as the data item and 1 as the length
6. In row 17, specify `line-feed` as the data item and `1` as the length.
7. Press [`Esc` + `E`] to save your changes and return to the command line.

Step 7 (Optional): Configuring CDR Format if CDR1 is Formatted For a Billing System

If CDR1 output is defined for your billing system and that billing system itself requires a customized format, then CDR1 and CDR2 require the same format. You cannot define a separate custom format for each CDR output when one format is defined for a billing system.

In this scenario, rather than editing the Primary Output fields for CDR2 as shown in "Step 5: Configuring CDR Endpoints and Format" on page 20, edit the following file so that its format matches that of the CDR1 formatted for the billing system:
`c:\Program Files\AppManager\bin\AvayaCM\AvayaCDRFormat.txt`

To reformat AvayaCDRFormat.txt:

1. Use the CDR System Parameters screen to verify the CDR format in use for the CDR1 assigned to the billing system. To access this screen, execute the `display system-parameters cdr` command, and then press [`Esc + N`] to navigate to the next page of the screen, which identifies field names and lengths.
2. Navigate to `c:\Program Files\AppManager\bin\AvayaCM` and open `AvayaCDRFormat.txt` in a text editor, such as Notepad.
3. Ensure the value of the `date-format` field in the file matches the value of the `CDR Date Format` field shown on the first page of the system-parameters cdr SAT screen, to which you navigated in Step **1**. The value of the `date-format` field should be either `month/day` or `day/month`.
4. Below the `date-format` field, delete and then replace all lines in the file with the fields that are configured in the customized CDR format in the system-parameters cdr SAT screen. Type all CDR fields in the same order and with the same field lengths as shown on the SAT screen. Type each field on a single line, with a space separating the field name from the field length.

   In order for the module to process CDRs, the customized CDR format must contain the following fields. The CDR format can contain either the `clg-num/in-tac` or the `calling-num` field, and either the `sec-dur` or the `duration` field.

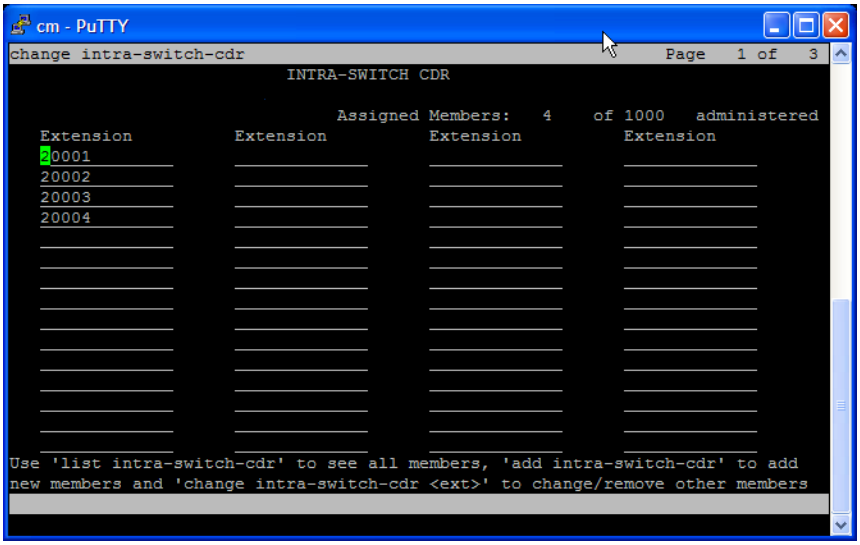
   - cond-code
   - cig-num/in-tac (or calling-num)
   - dialed-num
   - date
   - time
   - sec-dur (or duration)
5. Save `AvayaCDRFormat.txt`.

NOTE: If you uninstall the AppManager for Avaya Communication Manager module, the uninstallation process will remove `\AppManager\bin\AvayaCM\AvayaCDRFormat.txt` along with the module files. To preserve any edits you made to this file, copy it to another location before you begin the uninstallation process.

Step 8 (Optional): Specifying Intra-Switch Extension Numbers

If you enabled intra-switch CDRs in "Step 5: Configuring CDR Endpoints and Format" on page 20, specify the phone extension numbers for which you want intra-switch CDRs.

Use the Intra-Switch CDR screen to specify extension numbers. To access this screen, execute the following command: `change intra-switch-cdr`



In the Extension columns, type one extension number per line.

Press [`Esc` + `E`] to save your changes and return to the command line.

Updating the 46xxsettings.txt File

A setting on SIP (Session Initiation Protocol) phones identifies the location of the SIP phone configuration file. The setting points to a Web server, usually the SIP Enablement Server (SES).

On the Web server is the `46xxsettings.txt` file, which contains configuration settings to be sent to each SIP phone.

You should update the `46xxsettings.txt` file to identify the IP address of the proxy agent computer.

To update the 46xxsettings.txt file:

1. Open the file in a text editor.
2. Add the following line of text:

   `SET RTCPMON` *`IP address`*

   where *IP address* is the IP address of the proxy agent computer.
3. Reboot your SIP phones so that they pick up the new configuration setting.

Discovering Avaya Communication Manager Resources

Use the Discovery_AvayaCM Knowledge Script to discover Avaya Communication Manager configuration information and resources, including Switch Processing Elements (SPE), Enterprise Survivable Servers (ESS), Local Survivable Processors (LSP), H.248 media gateways, IP stations, attendant consoles, and remote office stations. You can also choose to discover NetIQ SNMP Trap Receiver. For more information, see "Working with NetIQ SNMP Trap Receiver" on page 105.

Only one Windows server can act as proxy for any given Communication Manager cluster. Therefore, run this script on only one computer at a time.

AppManager uses SNMP queries to access remote Communication Manager servers and to enable functionality of NetIQ SNMP Trap Receiver. Configure the SNMP community string information in AppManager Security Manager for each Communication Manager you want to monitor. The community string information allows AppManager to access the remote Communication Manager servers. For more information, see "Configuring SNMP Community Strings" on page 14.

NOTE: If you discover a call manager when you are bypassing SNMP (by selecting the **Discover using manual configuration?** parameter), the created treeview is restricted to the object types that the configuration supports. If the Knowledge Scripts that correspond to a particular object type are not supported for the configuration, they will not appear in the treeview. Review the Knowledge Script descriptions for specific restrictions.

Set the parameters on the **Values** tab as needed:

Parameter	How to Set It
General Settings	
Job Failure Notification	
Event severity when job fails	Set the severity level, from 1 to 40, to indicate the importance of the failure of the Discovery_AvayaCM job. The default is 5.
Set up supplemental database?	Select **Yes** to create the Avaya CM supplemental database, including the tables and stored procedures needed to store call detail records and phone deregistration information. The default is Yes. For more information, see "Understanding the Avaya CM Supplemental Database" on page 102.
Event severity when database setup fails	Set the event severity level, from 1 to 40, to indicate the importance of an event in which the Avaya CM supplemental database is not created. The default is 15. It is possible that the supplemental database was not created because the Discovery job ran on a computer on which SQL Server is not installed.
Number of days to keep call detail records	Specify the number of days' worth of CDRs to keep in the Avaya CM supplemental database. Data older than what you specify is discarded. The default is 7 days. **NOTE:** Run DeleteCDRRecords Knowledge Script to remove old records from the supplemental database.

Parameter	How to Set It
Local SQL Server Instance name	Specify the name of the local SQL Server instance (on the proxy computer) in which you want to create the new Avaya CM supplemental database. Leave this parameter blank to accept the default name.
Raise event if database setup succeeds?	Select **Yes** to raise an event if creation of the Avaya CM supplemental database is successful. The default is unselected.
Event severity when database setup succeeds	Set the event severity level, from 1 to 40, to indicate the importance of an event in which the Avaya CM supplemental database is created successfully. The default is 25.
SNMP	
Global SNMP Message timeout	Specify the number of seconds discovery should attempt an SNMP message request to an *individual* Communication Manager server before retrying the connection. The default is 120 seconds. The value you set here is the timeout value for *all* SNMP message requests for *all* AvayaCM Knowledge Script jobs.
Global SNMP Task timeout	Specify the number of seconds discovery should attempt an SNMP retrieve request to an *individual* Communication Manager server before retrying the connection. The default is 3600 seconds. The value you set here is the timeout value for *all* SNMP retrieve requests for *all* AvayaCM Knowledge Script jobs.
Global SNMP retries	Specify the number of times discovery should attempt an SNMP connection to an individual Communication Manager before attempting an SNMP connection to the next Communication Manager in the list. The default is 4 retries. The value you set here will be the number of retries for *all* SNMP connections for *all* AvayaCM Knowledge Script jobs. **Hint** If you experience timeouts that appear to be caused by lost messages rather than CPU usage, increase the number of retries, which affects SNMP `GETNext` and `GETBulk` requests. For example, if CPU is stable and you have already increased the timeout value, but packet loss in the network is high and timeouts are still being experienced, you can increase the number of retries.
Enable use of SNMP GETBulk requests during discovery?	By default, this parameter is enabled, allowing the Discovery_AvayaCM Knowledge Script job to use SNMP `GETNext` and `GETBulk` requests to access Communication Manager MIBs. Disable this parameter to allow the script to use only `GETNEXT` requests. Not all MIB tables are extensive enough to need a `GETBulk` request. A `GETBulk` request is faster, but more CPU-intensive than a `GETNext` request.
Number of rows to request for each GETBulk operation	Specify the number of rows from the MIB table to return in a `GETBulk` request. The default is 10 rows. The number of rows determines how quickly MIB data is returned. If CPU usage is too high, you can reduce the number of rows per `GETBulk` request or disable the *Enable use of SNMP GETBulk requests during discovery?* parameter.

Parameter	How to Set It
Interval to pause between GETBulk requests	Specify the number of milliseconds to wait between GETBulk requests. The default is 100 milliseconds. The delay can help manage CPU usage and speed of SNMP requests. For example, a one-row `GETBulk` with a 100-millisecond delay between requests executes slower and uses less CPU than a `GETNext` request.
Raise event if discovery succeeds?	Select **Yes** to raise an event if discovery succeeds in finding Communication Manager devices. The default is unselected.
Event severity when discovery succeeds	Set the severity level, from 1 to 40, to indicate the importance of an event in which discovery succeeds in finding Communication Manager devices. The default if 25.
Raise event if discovery fails?	Select **Yes** to raise an event if discovery fails to find some or all of your Communication Manager devices. The default is Yes.
Event severity when discovery fails	Set the severity level, from 1 to 40, to indicate the importance of an event in which discovery fails to find some or all of your Communication Manager devices. The default is 10.
Discover Avaya Communication Manager Servers	
Discovery timeout for all servers	Specify the number of minutes the script should attempt to discover *all* specified Communication Manager servers before stopping as unsuccessful. The maximum is 60 minutes. The default is 30 minutes.
Maximum number of concurrent discoveries	Specify the maximum number of Communication Manager servers that can be queried for discovery at one time. No matter what value you enter, discovery is still performed for the entire list of devices that you specify in the following parameters. Setting this parameter to a low value throttles the number of SNMP requests performed at one time, but may increase the overall time it takes to discover a list of devices. The default is 10 concurrent discoveries.

Parameter	How to Set It
Comma-separated list of active Communication Manager servers	Use this parameter if you know which Communication Manager servers you want to discover. Each duplex pair of AvayaCM call managers has an active server IP address that is shared between the duplex servers. This address is known as an IP-alias (or virtual address). The active server is the only server that responds on the IP-alias. Each server in the pair also responds to the physical address of that server. When a duplex call manager pair is discovered, the virtual address must be used rather than the physical address. If Domain Name Service (DNS) entries are mapped to the addresses and a DNS name is used for discovery, the DNS name used on a duplex pair should be the one mapped to the virtual address. **TIP:** You should discover the Communication Manager server by the virtual address, as a server discovered by physical address loses connectivity when the duplex processor changes the active status between the physical servers. Specify at least one IP address or hostname, using a comma to separate multiple items. For example: `10.0.1.1,10.0.1.7` You can enter IP addresses or hostnames, but you *must* enter the same IP address or hostname for which you configured SNMP community string information. If you configured a community string for a hostname, enter the *same* hostname; if you configured an IP address, enter the *same* IP address. For more information, see "Configuring SNMP Community Strings" on page 14.

Parameter	How to Set It

Comma-separated list of Communication Manager IP address pairs in a single NAT cluster	MSPs (Managed Service Providers) frequently maintain distributed customer networks in which NAT (Network Address Translation) is used to translate the IP address ranges that are monitored from a single NOC (Network Operations Center). The use of NAT prevents AppManager from recognizing the actual IP addresses of the servers in the remote cluster. If your AppManager agent is located on a server in the NOC, but the monitored devices are located in a cluster in the remote customer network, you need to provide AppManager with a list of the IP addresses of the remote monitored devices.
	
	Use this parameter to enable AppManager to recognize the IP addresses of the servers for a single remote Communication Manager cluster.
	
	Type a list of IP address pairs for the Communication Manager servers in a remote cluster. Use commas to separate the addresses. A pair consists of a server's NAT (external) IP address and its IP address inside the cluster. A Communication Manager cluster can contain three IP addresses: an active SPE virtual address, a primary physical address, and a secondary physical address. Each of these addresses must be represented by a pair in this parameter. A maximum of six IP addresses is allowed in this parameter.
	
	Use the following format:
	
	`externalactiveSPEvirtualaddress,internalactiveSPEvirtualaddress,externalprimaryphysicaladdress,internalprimaryphysicaladdress,externalsecondaryphysicaladdress,internalsecondaryphysicaladdress`
	
	In the following example, the 10.41* addresses are externally visible and the 172.16* addresses are visible only to the Communication Managers:
	
	`10.41.1.10,172.16.1.10,10.41.1.11,172.16.1.11,...`

Full path to file with list of active Communication Manager servers	Instead of listing each server separately, you can specify the full path to a file on the proxy computer that contains a list of IP addresses or hostnames of Communication Manager servers.
	
	In the file, specify the servers on multiple lines and ensure that each line contains only one entry. For example:
	
	`AvayaCM01`
	`AvayaCM02`
	`AvayaCM10`
	
	You can enter IP addresses or hostnames, but you *must* enter the same IP address or hostname for which you configured SNMP community string information. If you configured a community string for a hostname, enter the *same* hostname in the list; if you configured an IP address, enter the *same* IP address in your list.
	
	For more information, see "Configuring SNMP Community Strings" on page 14.

Parameter	How to Set It
Add Avaya Index to discovered names?	Select **Yes** to discover all Avaya objects for which indices will be collected.The collected indices will be placed in the treeview with the treeview name formed by suffixing the index of the object to the Avaya name of for the object. For example: “Fax test 1” at index 14 becomes `Fax test 1[14]` and “Fax test 2” at index 15 becomes `Fax test 2[15]`. The Avaya objects that use indices for identification include the following: ◆ Trunk Groups ◆ Gateways ◆ Hunt Groups If a trunk group has been named with a `num` suffix, do not use that suffix as a trunk group index. If there are two objects with the same name that have indices present, the names will be decomposed to identify the specific index so that the correct object is monitored. When this parameter is deselected, Avaya will suffix index numbers only to names that are not unique.
Resolve server names locally on agent?	Select **Yes** to override the Avaya call server MIB entries for server names with names found by the agent’s local name resolution lookup. The default is unselected. If set to Yes, the server name resolutions that are run on the agent will override the name values read through the Avaya call server MIB. The names displayed in the treeview will be the ones found by the agent when doing a reverse DNS for the discovered IP addresses. For example, s8000-001-EndCustomer1@MSP-com will display as s8000-001-EndCustomer1. The domain portion of the FQDN returned will not display. If set to Yes, and if the Network Address Translation (NAT) is in use (a list of address pairs entered for the NAT discovery parameter), the reverse DNS lookup will be for the addresses visible to the agent instead of the addresses for the call server side of the NAT. **NOTE:** If the set to Yes, an error case may occur if one or more of the server IP addresses (active SPE, server a or server b) cannot be resolved with a reverse DNS lookup because no lookup entry exists on either the DNS server or in the host file. If this occurs, the IP address that is used for the lookup is substituted for the missing name.
Discover using manual configuration	Select **Yes** to bypass SNMP and discover the data based on the IP address entered. Using a DNS lookup, the cluster name will be resolved, or if the lookup does not return a name, Avaya uses the IP address as the cluster name. Call data collection is supported for Call Managers discovered when bypassing SNMP, but Knowledge Scripts requiring SNMP are not supported. The treeview objects corresponding to SNMP-based Knowledge Scripts do not appear in the treeview. When this parameter is deselected, Avaya will use SNMP.
Discover Trap Receiver?	Select **Yes** to discover NetIQ SNMP Trap Receiver. When this parameter is selected, a treeview object is created for each call manager server (including ESS and LSP servers) in the cluster as well as for each H.248 gateway known to the cluster. You should configure the gateways to send traps directly to the Appmanager agent. For additional information about configuring gateway trap destinations, see https://downloads.avaya.com/css/P8/documents/100134724.The default is unselected.

Parameter	How to Set It
Trap Receiver IP address	Specify the IP address of the computer on which Trap Receiver is installed. The default is `localhost`.
Trap Receiver TCP port	Specify the TCP port number through which Trap Receiver will communicate with AppManager. The default is port 2735.
Configure Trap Receiver for associated servers?	Select Yes to allow Trap Receiver to listen for traps coming from other servers such as SIP Enablement Services (SES) servers or Application Enablement Services (AES) servers. The default is unselected.
Comma-separated list of associated server IP addresses	Provide a comma-separated list of the IP addresses of other servers. If you enabled the *Configure Trap Receiver for associated servers?* parameter, then Trap Receiver will listen for traps coming from the servers you specify.

Configuring Unique Port Numbers for Multiple Communication Managers

If you have not configured AppManager Security Manager with the port numbers needed for CDR and RTCP packet delivery, AppManager uses default port numbers:

- TCP port 9000 for CDR delivery
- UDP port 5005 for RTCP packet delivery

These port numbers are the same port numbers you configured on your Communication Manager in "Configuring Communication Manager To Send RTCP Packets and CDRs" on page 15. This process works well if you discovered only one Communication Manager per AppManager proxy agent computer.

If you discovered multiple Communication Managers, however, you must configure each Communication Manager to use unique port numbers for CDR and RTCP packet delivery. Use the instructions in "Configuring Communication Manager To Send RTCP Packets and CDRs" on page 15 for this purpose. Then, configure Security Manager to recognize the new port numbers.

NOTE: AppManager uses the port numbers to determine which CDRs and RTCP packets belong to a given Communication Manager. If you do not configure multiple Communication Managers to use unique port numbers, AppManager cannot determine which Communication Manager is associated with the gathered data.

On the Custom tab in Security Manager, complete the following fields:

Field	Description
Label	`AvayaCM_CallDataCollection`
Sub-label	IP address or hostname of the Communication Manager server. The IP address or hostname you type here must match the address or hostname you provide in the Discovery_AvayaCM Knowledge Script. For more information, see "Discovering Avaya Communication Manager Resources" on page 26.
Value 1	TCP port number that will be used for CDR delivery. This number must match the number you specified in "Step 4: Defining the CDR Link to the Proxy Agent Computer" on page 19.

Field	Description
Value 2	UDP port number that will be used for RTCP packet delivery. This number must match the number you specified in "Step 1: Configuring IP Address and Port for RTCP Packets" on page 16.
Value 3	Frequency with which the AppManager agent on the proxy computer will listen for RTCP packets sent from Avaya phones. Valid values are between 5 and 30. This value must match the value you specified in "Step 1: Configuring IP Address and Port for RTCP Packets" on page 16. For information about what happens when the values do not match, see "PhoneQuality Script Not Collecting Complete Data" on page 37.

Upgrading Knowledge Script Jobs

If you are using AppManager 8.x or later, the module upgrade process now *retains* any changes you may have made to the parameter settings for the Knowledge Scripts in the previous version of this module. Before AppManager 8.x, the module upgrade process *overwrote* any settings you may have made, changing the settings back to the module defaults.

As a result, if this module includes any changes to the default values for any Knowledge Script parameter, the module upgrade process ignores those changes and retains all parameter values that you updated. Unless you review the management guide or the online Help for that Knowledge Script, you will not know about any changes to default parameter values that came with this release.

You can push the changes for updated scripts to running Knowledge Script jobs in one of the following ways:

- Use the AMAdmin_UpgradeJobs Knowledge Script.
- Use the Properties Propagation feature.

Running AMAdmin_UpgradeJobs

The AMAdmin_UpgradeJobs Knowledge Script can push changes to running Knowledge Script jobs. Your AppManager repository (`QDB`) must be at version 7.0 or later. In addition, the repository computer must have hotfix 72040 installed, or the most recent AppManager repository hotfix. To download the hotfix, see the AppManager Suite Hotfixes page.

Upgrading jobs to use the most recent script version allows the jobs to take advantage of the latest script logic while maintaining existing parameter values for the job.

For more information, see the **Help** for the AMAdmin_UpgradeJobs Knowledge Script.

Propagating Knowledge Script Changes

You can propagate script changes to jobs that are running and to Knowledge Script Groups, including recommended Knowledge Script Groups and renamed Knowledge Scripts.

Before propagating script changes, verify that the script parameters are set to your specifications. You might need to appropriately set new parameters for your environment or application.

If you are not using AppManager 8.x or later, customized script parameters might have reverted to default parameters during the installation of the module.

You can choose to propagate only properties (specified in the **Schedule** and **Values** tabs), only the script (which is the logic of the Knowledge Script), or both. Unless you know specifically that changes affect only the script logic, you should propagate the properties and the script.

For more information about propagating Knowledge Script changes, see the "Running Monitoring Jobs" chapter of the *Control Center User Guide for AppManager*.

Propagating Changes to Ad Hoc Jobs or Knowledge Script Groups

You can propagate the properties and the logic (script) of a Knowledge Script to ad hoc jobs started by that Knowledge Script. Corresponding jobs are stopped and restarted with the Knowledge Script changes.

You can also propagate the properties and logic of a Knowledge Script to corresponding Knowledge Script Group members. After you propagate script changes to Knowledge Script Group members, you can propagate the updated Knowledge Script Group members to associated running jobs. Any monitoring jobs started by a Knowledge Script Group member are restarted with the job properties of the Knowledge Script Group member.

To propagate changes to ad hoc Knowledge Script jobs or Knowledge Script Groups:

1. In the Knowledge Script view, select the Knowledge Script or Knowledge Script Group for which you want to propagate changes.
2. Right-click the script or group and select **Properties propagation** > **Ad Hoc Jobs**.
3. Select the components of the Knowledge Script that you want to propagate to associated ad hoc jobs or groups and click **OK**:

Select	To propagate
Script	The logic of the Knowledge Script.
Properties	Values from the Knowledge Script Schedule and Values tabs, such as schedule, monitoring values, actions, and advanced options. If you are using AppManager 8.x or later, the module upgrade process now *retains* any changes you might have made to the parameter settings for the Knowledge Scripts in the previous version of this module.

Setting Up MSDE

Microsoft SQL Server 2000 Desktop Engine (MSDE) is a reduced version of Microsoft SQL Server. AppManager for Avaya Communication Manager supports the use of MSDE Service Pack 4 in limited environments, such as those with only one Communication Manager.

Before you can create the Avaya CM supplemental database, install or configure MSDE if you do not use a full version of Microsoft SQL Server. For more information, see "Understanding the Avaya CM Supplemental Database" on page 102.

To set up MSDE for use with AppManager for Avaya Communication Manager:

1. Download the MSDE package from the Microsoft Web site:

   www.microsoft.com/downloads/details.aspx?FamilyID=8e2dfc8d-c20e-4446-99a9-b7f0213f8bc5&DisplayLang=en

Microsoft SQL Server 2000 Service Pack 4 SP4 is a cumulative service pack containing all of the fixes from previous service packs, including fixes for SQL Server 2000 Desktop Engine (MSDE) components.

2. Follow the instructions on the download page to save `SQL2000.MSDE-KB884525-SP4-x86.EXE` to disk. After saving it, run it from the disk and accept the license agreement. By default, the package extracts itself to the `C:\SQL2KSP4` folder.
3. Open a command prompt window and navigate to `C:\SQL2KSP4`.
4. You use command-line arguments to install MSDE. See the `ReadmeSql2k32Sp4.htm` file for explanations of the various options. The following is just one possibility:

   ```
   setup SAPWD=mypassword SECURITYMODE=SQL /L*v c:/MSDELog.log
   ```

   The preceding argument installs MSDE to the default instance, sets up mixed mode security, and sets the sa password. The `/L*v c:/MSDELog.log` at the end causes a verbose installation log to be written to the `c:\MSDELog.log` file. Note the forward slashes used in the filename when specifying it on the command line.

   The MSDE installation does not automatically start the `MSSQLSERVER` service.

   NOTE: By default, this installation of MSDE does not enable network protocols, which means you cannot connect to it remotely. According to the MSDE documentation, you can specify `DISABLENETWORKPROTOCOL=0` on the command line to enable network protocols. You can also use the SQL Server network configuration utility.

   ---

5. From the Control Panel, double-click **Administrative Tools**, and then double-click **Services**. Confirm that the `MSSQLSERVER` service is set for Automatic startup. Set the `SQLSERVERAGENT` service to Automatic startup.
6. On the desktop, right-click **My Computer** and select **Properties**. In the System Properties dialog box, click the **Advanced** tab and then click **Environment Variables**. Confirm that `C:\Program Files\Microsoft SQL Server\80\Tools\Binn` is in the PATH environment variable. It should have been added to the path by the MSDE installation. If not, add it yourself.
7. To enable network protocols, navigate to the `C:\Program Files\Microsoft SQL Server\80\Tools\Binn` folder and run `svrnetcn.exe`.
8. Move the **TCP/IP** protocol from the **Disabled** list to the **Enabled** list and click **OK**.
9. Reboot the computer to ensure the PATH environment variable is propagated to all services. Rebooting also automatically starts up the `MSSQLSERVER` and `SQLSERVERAGENT` services.

Understanding the Log Configuration File

AppManager for Avaya Communication Manager uses a logging mechanism that differs from that of other AppManager modules. When installed, the module lays down a log configuration file, `log4cxx.cfg`, that you can use to control the size, content, and number of backups for the `AvayaCM.log` file.

The configuration file is saved by default to `C:\Program Files\NetIQ\AppManager\bin`. From here, you can edit `log4cxx.cfg` to make the following changes:

Line of Code	Possible Changes

log4j.rootLogger=	This line of code controls the types of events that are saved to the `AvayaCM.log` file.
	
	Type the following to log all events:
	`log4j.rootLogger=Debug, AvayaCM`
	
	Type the following to log only informational events:
	`log4j.rootLogger=Info, AvayaCM`
	
	Type the following to log only warning events:
	`log4j.rootLogger=Warn, AvayaCM`
	
	Type the following to log only error events:
	`log4j.rootLogger=Error, AvayaCM`

log4j.appender.AvayaCM.MaxFileSize=	This line of code controls the maximum size to which the `AvayaCM.log` file can grow. By default, the maximum size is 10MB. For example, to change the maximum size to 100MB, type the following: `log4j.appender.AvayaCM.MaxFileSize=100MB`
log4j.appender.AvayaCM.MaxBackupIndex=	This line of code controls the number of backup copies of `AvayaCM.log` created when the log reaches its maximum size. The default is 1, which means when `AvayaCM.log` reaches its maximum size, the full log is saved and a new log is created. When the new log is full, the process is repeated, and the old backup copy is deleted. To create more than one backup copy, change the value of this line of code. For example, to create three backup copies, type the following: `log4j.appender.AvayaCM.MaxBackupIndex=3`

Troubleshooting

This topic provides answers to problems you may encounter when monitoring Avaya Communication Manager with AppManager.

Port Number Already in Use

Problem: You receive an event message similar to the following:

```
UDP port <port number> is already in use by phones associated with the
Communication Manager at address <IP address>. You must specify a different UDP
port for RTCP packets to be sent to/from the Communication Manager at address <IP
address>.
```

Possible Cause: You discovered more than one Communication Manager. Events are raised if multiple Communication Managers compete for the same port number. AppManager is configured to use UDP port 5005 for RTCP packet delivery and TCP port 9000 for CDR delivery.

Solution: Configure your Communication Managers to use unique port numbers. Then configure AppManager Security Manager with the new port numbers. For more information, see "Configuring Communication Manager To Send RTCP Packets and CDRs" on page 15 and "Configuring Unique Port Numbers for Multiple Communication Managers" on page 32.

PhoneQuality Script Not Collecting Complete Data

Problem: The PhoneQuality script does not gather data about phone calls you know took place.

Possible Cause: There is a large difference between the frequency with which Communication Manager sends RTCP packets to AppManager and the frequency with which AppManager listens for incoming RTCP packets. If the frequencies are different, some calls may be lost:

- When RTCP packets are sent less frequently than AppManager listens for them, AppManager may determine a call is completed when it is not. AppManager then marks the call complete and discards any remaining packets that arrive later for that call.
- When RTCP packets are sent more frequently than AppManager listens for them, it takes AppManager longer to determine a call is completed, especially when the call does not contain a `BYE` message, which happens occasionally.

Solution: Configure Communication Manager to send RTCP packets at the same frequency with which AppManager listens for them. For more information, see "Configuring Communication Manager To Send RTCP Packets and CDRs" on page 15 and "Configuring Unique Port Numbers for Multiple Communication Managers" on page 32.

PhoneQuality Data Points Returned Inconsistently

Problem: The PhoneQuality script returns data points at inconsistent intervals.

Possible Cause: If you monitor a large number of active phones and you collect data for most or all monitored statistics (MOS, R-Value, jitter, latency, packet loss), the AppManager agent may be unable to keep pace with the amount of data that is being collected. In this situation, the AppManager agent throttles any collected data, and then, when processing capacity permits, creates one data point based on an average of the throttled values. This throttling can result in data points created at various, inconsistent times during the monitoring period.

Solution: Perform one or both of the following tasks:

- Reduce the number of active phones you are monitoring
- Collect data for fewer statistics

Discovery Returns “SNMP Request Failure” Error

Problem: Discovery returns an `SNMP Request Failure` error message.

Possible Cause: A network router or switch exists between your Communication Manager servers and the proxy agent computer, and the Communication Servers are unable to recognize the IP address of the proxy agent computer. Instead, they see only the IP address of the router or switch, and so are unable to connect to the proxy agent computer using SNMP.

Solution: Configure Communication Manager to recognize the IP address of the router or switch and the IP address of the proxy computer.

To configure the IP address:

1. Navigate to the Integrated Management Maintenance Web page for your Communication Manager.
2. For **Community name**, type `ion-test`.
3. In the left pane, click **SNMP Agents**.
4. In the “IP Addresses for SNMP Access” section, select **Following IP addresses**.
5. In the **IP address** fields, type the IP addresses of your proxy agent computer and the router or switch that lies between the proxy agent computer and the Communication Manager server.
6. Click **Submit**.

Knowledge Script Returns “SNMP Timeout” Error

Problem: Running an AvayaCM Knowledge Script job returns the following error:
`CHR0392: An SNMP request sent to [IP address] timed out.`

Possible Causes: An SNMP timeout has several possible causes:

- The Avaya SNMP agent is set to SNMP version 1.
- The server is down or is not running an SNMP agent.
- The server’s SNMP agent is down.
- Network congestion or packet loss occurred during the SNMP request.
- The SNMP community string you provided is incorrect.
- The queried `G3-AVAYA-MIB` SNMP tables have not repopulated.
- The default SNMP timeout period is too short.
- The Communication Manager is experiencing an internal failure that prevents SNMP communication.

Solutions: Try one or more of the following solutions:

- Configure *each* server for SNMP v2. For more information, see “Administering SNMP Agents” in the *Administrator Guide for Avaya Communication Manager*. If you use the Integrated Management Maintenance Web page to configure SNMP settings in a cluster, the SNMP settings apply only to the server that you are connected to and not to all servers in the cluster. Connect to *each* server individually and configure SNMP and the firewall.
- Verify the status of the server or the existence of the SNMP agent.
- Verify the status of the SNMP agent.
- Wait and run the script later. Or, if you have NetIQ Vivinet Diagnostics installed, run a diagnosis on the affected IP address.
- Provide correct community strings in AppManager Security Manager. For more information, see “Configuring SNMP Community Strings” on page 14.
- Wait for tables to be repopulated. Some tables in the Avaya MIBs repopulate only once an hour.
- Provide a longer timeout value in the *Global SNMP timeout* parameter in the Discovery_AvayaCM script. You can verify the current timeout period by selecting the Active SPE object in the TreeView pane and clicking the Details tab. After changing the timeout value, rerun the Discovery job.

Supplemental Database Grows Too Large

Problem: The AvayaCM supplemental database is not pruned by the SQL job that runs every night to clean up the database.

Possible Cause: Your supplemental database is on a SQL Server 2005 Express computer. The overnight SQL job requires Integration Services, which is not supported on SQL Server 2005 Express.

Solution: Run the following stored procedure from a command line. Then configure a Windows Scheduled Task to schedule pruning at an interval of your choosing.

```
osql -E -S <sql server> -n -d <
database
> -" "exec dbo.Task_AvayaCM_Pruni"ng
```

where <*sql server*> is the name of the server that hosts the supplemental database, and where <*database*> is the name of the database.

For example: `osql -E -S SuppDBAvaya -n -d AvayaCM_S8300-Cluster -Q "exec dbo.Task_AvayaCM_Pruning”`

The process for configuring a Windows Scheduled Task varies according to your version of Microsoft Windows. Consult your Windows documentation for more information.

2 Introducing AppManager for Avaya Communication Manager

Avaya Communication Manager (also known as Avaya Aura Communication Manager) is a scalable, survivable telephony solution for enterprises and branch offices, providing call processing, messaging, and contact center functions.

This chapter introduces AppManager for Avaya Communication Manager, providing an overview of the module and describing how you can use AppManager to better monitor Communication Manager clusters, including vital components as Switch Processing Elements (SPE), Enterprise Survivable Servers (ESS), Local Survivable Processors (LSP), H.248 media gateways, IP stations, attendant consoles, remote office stations, hunt groups, trunk groups, and announcement ports.

Features and Benefits

The following are just a few of the features and benefits of monitoring Avaya Communication Manager with AppManager:

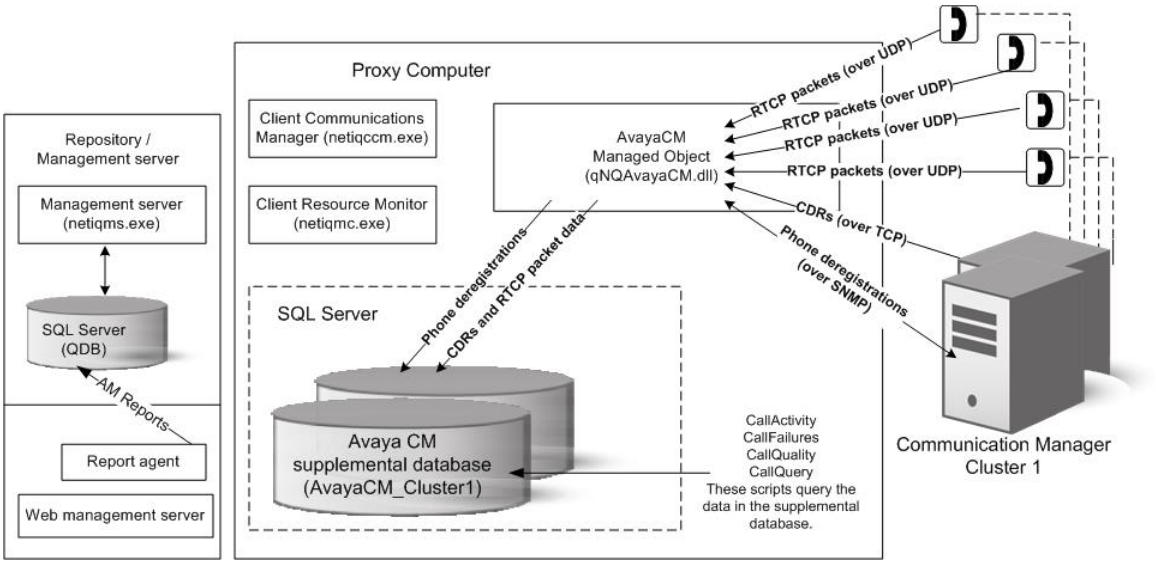
- Discover Communication Manager clusters with a single discovery job
- Diagnose call and phone quality using NetIQ Vivinet Diagnostics. For more information, see "Triggering Call and Phone Quality Diagnoses" on page 66.
- Use Knowledge Scripts to collect data for all monitored Communication Managers and associated components:
- Call activity metrics such as active and completed calls
- Call quality metrics such as jitter, delay, lost data, and MOS
- Call failures
- CPU usage and available CPU
- Phone deregistration and disconnection status
- Real-time voice quality statistics on active phones
- Historical call activity
- Inventory of phones configured for a Communication Manager
- H.248 gateway metrics such as major, minor, and warning alarms, and H.248 link availability
- Trunk group metrics such as busy time, calls in and out of queue, and out-of-service trunks
- Server uptime
- Hunt group metrics such as answered calls, queued calls, abandoned calls, and call wait time
- ESS and LSP registration status
- Announcement activity such as dropped calls, queued calls, and peak port usage
- Security violations such as barrier code violations, calls that generated authorization code violations, and calls that generated station security code violations
- SNMP traps forwarded from NetIQ SNMP Trap Receiver

Proxy Architecture and Deployment

With AppManager support for Avaya Communication Manager, the agent does not need to be installed on every device you want to monitor. With this *proxy* architecture, the module is installed on a proxy agent computer. When you run a Knowledge Script job, the managed object runs on the proxy computer and uses SNMP to send messages to and from Communication Manager.

Unlike other AppManager modules, AppManager for Avaya Communication Manager supports only one AppManager repository (`QDB`) per proxy computer. This limitation ensures the accuracy of monitoring phones with the PhoneQuality script. The list of phones available for monitoring with the PhoneQuality script does not differentiate between repositories; if multiple repositories were allowed, you could very well monitor the wrong set of phones for a given repository.

The following diagram illustrates the flow of data between the components of AppManager for Avaya Communication Manager:



Avaya Communication Manager uses TCP to send call detail records (CDRs) to the managed object. Phones registered to Communication Manager use RTCP to send call packets to the managed object. The managed object sends the CDRs and RTCP packets to the Avaya CM supplemental database while the CallActivity, CallFailures, and CallQuality Knowledge Scripts are running. The managed object uses SNMP queries to ask Communication Manager which phones are registered, and then sends deregistration information to the supplemental database while the PhoneDeregistration Knowledge Script is running.

The Knowledge Scripts monitor and raise events for the data in the supplemental database according to the parameters you set in the scripts. For more information, see “SetupSupplementalDB” on page 101.

Counting AppManager Licenses

The AppManager for Avaya Communication Manager module consumes one AppManager license for every registered IP station.

When used with Avaya call managers that support SNMP, the Avaya Communication Manager applies one AppManager license for every registered IP station. When used with Avaya call managers that are not discovered using SNMP, the Avaya Communication Manager applies one AppManager license for every IP subscriber as estimated by the unique extensions and endpoint IP data appearing in the CDR data collected.

3 AvayaCM Knowledge Scripts

AppManager provides the following Knowledge Scripts for monitoring Avaya Communication Manager servers and resources.

From the Knowledge Script view of Control Center, you can access more information about any NetIQ-supported Knowledge Script by selecting it and clicking **Help**. In the Operator Console, click any Knowledge Script in the Knowledge Script pane and press **F1**.

Knowledge Script	What It Does
AddMIB	Adds management information bases for monitoring by the SNMPTrap Knowledge Script.
AddPhone	Adds Avaya IP phones as objects in the TreeView pane.
Announcements	Monitors announcements for queued calls, dropped calls, and peak port usage.
AttendantCalls	Monitors the switch processing element (SPE) for answered calls, abandoned calls, calls on hold, queued calls, active time, and average answer time.
CallActivity	Monitors call activity on selected Communication Managers.
CallFailures	Monitors calls for abnormal causes of termination.
CallQuality	Monitors calls for quality metrics such as jitter, latency, packet loss, MOS (Mean Opinion Score), and R-Value.
CallQuery	Queries call detail records retrieved from Communication Manager and stored in the Avaya CM supplemental database.
CPU_Usage	Monitors a Communication Manager server for system management CPU usage, operating system CPU usage, call processing CPU usage, and available CPU.
DeleteCDRRecords	Deletes call detail records (CDR) from supplemental database.
ESS_Status	Monitors the registration status of an Enterprise Survivable Server (ESS).
H248GatewayStatus	Monitors H.248 media gateways for alarms and unavailable H.248 links.
HuntGroupUsage	Monitors hunt groups for answered calls, queued calls, abandoned calls, and average call wait time.
LSP_Status	Monitors the registration status of Local Survivable Processors (LSP).
PhoneConnectivity	Monitors disconnected registered phones for a Communication Manager and retains a history of the monitored phones in the Avaya CM supplemental database.
PhoneDeregistrations	Monitors phone deregistrations for a Communication Manager and retains a history of the monitored phones in the Avaya CM supplemental database.

Knowledge Script	What It Does
PhoneInventory	Creates an inventory of the phones configured in a Communication Manager cluster.
PhoneQuality	Collects real-time voice quality statistics for active calls on Avaya IP phones.
RegisteredResources	Monitors changes in the number of resources registered on a Communication Manager server.
RemovePhone	Removes Avaya IP phone objects from the TreeView pane.
RetrieveConfigData	Retrieves Communication Manager configuration data about stations and gateways and stores it in the Avaya CM supplemental database.
SecurityViolations	Monitors security violations for barrier codes, monitors calls that generated authorization code violations, and monitors calls that generated station security violations.
SetupSupplementalDB	Creates an Avaya CM supplemental database in which to store call detail records, disconnected phone information, and deregistered phone information.
SNMPTrap	Checks for incoming SNMP traps forwarded from NetIQ SNMP Trap Receiver.
SystemUptime	Monitors the number of hours a Communication Manager has been operational since its last reboot.
TrunkGroupUsage	Monitors trunk groups for busy time, calls queued and not queued, and out-of-service trunks.
Recommended Knowledge Script Group	Runs all recommended Knowledge Scripts at one time.

AddMIB

Use this Knowledge Script to add MIB (management information base) files to the MIB tree that is monitored by the SNMPTrap Knowledge Script. The MIB files should be `ASN.1` text files with `.txt` or `.my` file extensions. The MIB files should not be compiled MIB files.

With this script you can copy a MIB file from an arbitrary directory to the MIB tree located in the `AppManager directory\bin\MIBs` directory. And, by using the `Reload MIB tree?` parameter, you can reload all MIBs in the tree without restarting the AppManager agent. A restart of the AppManager agent automatically reloads the MIB tree.

Scenarios for using this script include the following examples:

In This Scenario	Set These Parameters
You want to add a MIB file to the MIB tree, but do not want the addition to take effect until after the next restart of the AppManager agent.	*Full path to MIB files* and *List of MIB files*: Provide location and name of MIB file you want to add. *Reload MIB tree?*: Set to **No** (unselected).

In This Scenario	Set These Parameters
You manually copied a MIB file to the MIB directory and want to reload all MIBs in the directory.	*Full path to MIB files* and *List of MIB files*: Leave blank. *Reload MIB tree?*: Select **Yes**. *MIB reload timeout*: Set new timeout value or accept default of 10 seconds.
To fix compiler errors, you edited some MIBs in the MIB directory. Now you want to reload the MIBs to ensure the errors have been fixed.	*Full path to MIB files* and *List of MIB files*: Leave blank. *Reload MIB tree?*: Select **Yes**. *MIB reload timeout*: Set new timeout value or accept default of 10 seconds.

Resource Object

AvayaCM Trap Receiver object

Default Schedule

By default, this script runs once.

Setting Parameter Values

Set the following parameters as needed.

Parameter	How to Set It
Full path to MIB files	Specify the full path to the folder that contains the MIB files you want to install. The AppManager agent on the proxy agent computer must have network access to the location you specify.
List of MIB files	Provide a comma-separated list of the MIB files you want to install. The MIB files should be `ASN.1` text files with `.txt` or `.my` file extensions. The MIB files should not be compiled MIB files. The MIB files you specify must be located in the folder you identified in the *Full path to MIB files* parameter.
Reload MIB tree?	Select **Yes** to update the MIB tree.
MIB reload timeout	Specify the length of time AppManager should attempt to update the MIB tree before timing out and raising a failure event. The default is 10 seconds.
Event Notification	
Raise event if installation and reloading of MIB tree succeeds?	Select **Yes** to raise an event if installation of the MIB files and reloading of the MIB tree succeeds. The default is Yes. **NOTE:** Reloading of the MIB tree can be successful even if no new MIB files are installed. Reloading of the MIB tree can proceed even if you provide no MIB files in the *List of MIB files* or *Full path to list of MIB files* parameter.

Parameter	How to Set It
Event severity when installation and reloading of MIB tree succeeds	Set the severity level, from 1 to 40, to indicate the importance of an event in which the installation of MIB files and the reloading of the MIB tree succeeds. The default is 25.
Raise event if reload MIB parser warnings received?	Select **Yes** to raise an event if warning messages are received during the reload process. The default is Yes. Warning scenarios include: ◆ MIBs are installed successfully but the *Reload MIB tree?* parameter is not set to Yes. ◆ Not all specified MIB files were loaded to the MIB tree.
Event severity when reload MIB parser warnings received	Set the severity level, from 1 to 40, to indicate the importance of an event in which warning messages are received during the reload process. The default is 15.
Raise event if installation and reloading of MIB tree fails?	Select **Yes** to raise an event if AppManager fails to install or reload the specified MIB files. The default is Yes. Failure scenarios include: ◆ MIB reload timeout period expired. ◆ Not all specified MIB files were installed.
Event severity when installation and reloading of MIB tree fails	Set the severity level, from 1 to 40, to indicate the importance of an event in which the installation or reloading of the MIB tree fails. The default is 10.
Raise event with the list of currently installed MIBs?	Select **Yes** to raise an informational event that provides a list of all MIBs installed in the MIB tree. The default is Yes.
Event severity for the list of currently installed MIBs	Set the severity level, from 1 to 40, to indicate the importance of an event that provides a list of all MIBs installed in the MIB tree. The default is 25.

AddPhone

Use this Knowledge Script to add Avaya IP phones as objects to be monitored. You must add a phone before you can monitor it with the PhoneQuality script. This script raises an event when phones are added or if phones cannot be added.

Resource Object

AvayaCM Station folder

Default Schedule

By default, this script runs once.

Setting Parameter Values

Set the following parameters as needed:

Parameter	How to Set It
General Settings	
Job Failure Notification	
Event severity when job fails	Set the event severity level, from 1 to 40, to indicate the importance of the failure of the AddPhone job. The default is 5.
Retrieve SNMP configuration data for these phones?	Select **Yes** to retrieve SNMP configuration data for these phones. This parameter is deselected by default. If you select **Yes**, you can run the addPhone script without first running the RetrieveConfigData script. If you select this parameter for a call manager that does not support SNMP data collection (indicated by the station folder details which display “no” for the SNMP capability), Avaya ignores the selection. If you do not select **Yes**, you must run RetreiveConfigData before running AddPhone.
Configuration Settings	
List of phone extensions	Provide a list of the extension numbers of the Avaya phones that you want to monitor. You can type one extension, a list of extensions, or a list of extension ranges. Separate multiple extensions with a comma. For example: `20001-20040,30001-30050,40000` **NOTE:** If you have many extension numbers, you can list the extensions in a separate file and then use the following parameter to access that file.

Full path to file with list of phone extensions	If you have many extensions to monitor, you can type the full path to a file that contains a list of the phone extensions. Each extension or range of extensions in the file should be on a separate line. For example:
	`20001-20040`
	`30001-30050`
	`40000`
	Because the file must be accessible from the AppManager agent, the path must be a local directory on the proxy computer or a UNC path.
	Important If you type a UNC path, then the `netiqmc` service must be running as a user that has access to the path.

Event Notification	
Raise event if all phones are added successfully?	Select **Yes** to raise an event if the specified phones are successfully added to the TreeView pane. The default is Yes.
Event severity when phones are added successfully	Set the event severity level, from 1 to 40, to indicate the importance of an event in which the specified phones are added successfully. The default is 25.
Raise event if configuration retrieval succeeds?	Select **Yes** to raise an event if the Knowledge Script successfully retrieved SNMP configuration information for these phones. The default is unselected.
Event severity when the configuration retrieval succeeds	Set the event severity level, from 1 to 40, to indicate the importance of an event in which the Knowledge Script successfully retrieved SNMP configuration information for these phones. The default is 25.

Announcements

Use this Knowledge Script to monitor announcements for queued calls, calls that dropped while in queue, and peak usage of announcement ports.

Choose whether to monitor specific announcements or the top *n* announcements. This script raises an event if a threshold is exceeded. In addition, this script generates data streams for calls queued, calls dropped, and peak port usage.

NOTE: If you bypass SNMP to discover a call manager (by selecting the **Discover using manual configuration** parameter), this script is not available.

Resource Object

AvayaCM Announcement object

Default Schedule

By default, this script runs every hour.

Setting Parameter Values

Set the following parameters as needed:

Parameter	How to Set It
General Settings	
Job Failure Notification	
Event severity when job fails	Set the event severity level, from 1 to 40, to indicate the importance of the failure of the Announcements job. The default is 5.
Select monitor type	Select one of the following monitoring options: ◆ **Top-N** — select to monitor *n* announcements with the highest values for queued calls, dropped calls, or peak port usage. If you select this option, provide the value for *n* in the *Number of announcements to monitor* parameter. ◆ **Comma-separated** — select to monitor specific announcements. If you select this option, provide a list of announcement extensions in the *Comma-separated list of announcements to monitor* parameter.
Number of announcements to monitor	Specify the number of announcements you want to monitor. The default is 5.
Comma-separated list of announcement extensions	Provide a list of the announcement extensions you want to monitor. You can provide individual extension numbers, a range of extension numbers, or a combination of both. For example: `20001-20020,20055,20100-20200` Separate each number or range with a comma.

Parameter	How to Set It
Enable use of SNMP GETBulk operations?	Select **Yes** to allow this Knowledge Script to access Communication Manager MIBs using `GETNext` and `GETBulk` SNMP requests, as appropriate. The default is Yes. Deselect the check box to allow the Knowledge Script to use only `GETNext` requests. Not all MIB tables are extensive enough to need a `GETBulk` request. A `GETBulk` request is faster, but more CPU-intensive than a `GETNext` request.
Number of rows to request for each GETBulk operation	Specify the number of rows from the MIB table to return in a `GETBulk` request. The default is 10 rows. The number of rows determines how fast the MIB data is returned. If the CPU usage is too high, you can reduce the number of rows per `GETBulk` request or deselect the *Enable use of SNMP GETBulk requests?* parameter.
Interval to pause between GETBulk operations	Specify the number of milliseconds to wait between GETBulk requests. The default is 100 milliseconds. The length of the delay can help manage CPU usage and the speed of SNMP requests. For example, a one-row `GETBUlk` with a 100-millisecond delay between requests executes more slowly and uses less CPU than a `GETNext` request.
Monitor Calls Queued	
Event Notification	
Raise event if number of calls queued exceeds threshold?	Select **Yes** to raise an event if the number of calls in queue for an announcement exceeds the threshold you set. The default is Yes.
Threshold - Maximum calls queued	Specify the highest number of calls that can be in queue for an announcement before an event is raised. The default is 0 calls.
Event severity when number of calls queued exceeds threshold	Set the event severity level, from 1 to 40, to indicate the importance of an event in which the number of calls in queue for an announcement exceeds the threshold. The default is 15.
Data Collection	
Collect data for calls queued?	Select **Yes** to collect data for charts and reports. If enabled, data collection returns the number of calls in queue for an announcement during the monitoring period. The default is Yes.
Monitor Calls Dropped	
Event Notification	
Raise event if number of calls dropped exceeds threshold?	Select **Yes** to raise an event if the number of calls dropped while in queue exceeds the threshold you set. The default is Yes.
Threshold - Maximum calls dropped	Specify the highest number of calls that can be dropped while in queue before an event is raised. The default is 0 calls.
Event severity when number of calls dropped exceeds threshold	Set the event severity level, from 1 to 40, to indicate the importance of an event in which the number of calls dropped while in queue exceeds the threshold. The default is 15.

Parameter	How to Set It
Data Collection	
Collect data for calls dropped?	Select **Yes** to collect data for charts and reports. If enabled, data collection returns the number of calls dropped while in queue during the monitoring period. The default is unselected.
Monitor Peak Ports Used	
Event Notification	
Raise event if peak number of ports used exceeds threshold?	Select **Yes** to raise an event if the number of ports in use simultaneously exceeds the threshold you set. The default is Yes.
Threshold - Maximum peak ports used	Specify the highest number of ports that can be in use simultaneously before an event is raised. The default is 12 ports.
Event severity when peak number of ports used exceeds threshold	Set the event severity level, from 1 to 40, to indicate the importance of an event in which the number of ports in use simultaneously exceeds the threshold. The default is 15.
Data Collection	
Collect data for peak ports used?	Select **Yes** to collect data for charts and reports. If enabled, data collection returns the highest number of ports in use simultaneously during the monitoring period. The default is Yes.

AttendantCalls

Use this Knowledge Script to monitor an active switch processing element (SPE) for statistics related to call attendants. This script raises an event if a monitored value exceeds the threshold you set. In addition, this script generates data streams for the following statistics:

- Answered calls
- Calls abandoned before being answered
- Calls abandoned while on hold
- Calls placed on hold
- Queued calls
- Number of minutes attendants are active
- Average call answer time

NOTE: If you discover a call manager with `AVAYA-AURA-CM-MIB SNMP` MIBs (Communication Manager version 6.3.141 or later), or if you bypass SNMP to discover a call manager, this script is not available.

Resource Object

AvayaCM Active SPE object

Default Schedule

By default, this script runs every hour because the SNMP data it monitors is updated only once an hour. If you change the schedule to a shorter interval, you may receive SNMP request errors until the SNMP data is repopulated.

Setting Parameter Values

Set the following parameters as needed:

Parameter	How to Set It
General Settings	
Job Failure Notification	
Event severity when job fails	Set the event severity level, from 1 to 40, to indicate the importance of the failure of the AttendantCalls job. The default is 5.
Monitor Answered Calls	
Event Notification	
Raise event if number of answered calls exceeds threshold?	Select **Yes** to raise an event if the number of calls answered by attendants exceeds the threshold you set. The default is Yes.
Threshold - Maximum number of answered calls	Specify the maximum number of calls that can be answered by attendants before an event is raised. The default is 100 calls.
Event severity when number of answered calls exceeds threshold	Set the severity level, from 1 to 40, to indicate the importance of an event in which the number of answered calls exceeds the threshold. The default is 15.
Data Collection	
Collect data for number of answered calls?	Select **Yes** to collect data for charts and reports. If enabled, data collection returns the number of calls answered by attendants during the monitoring period. The default is Yes.
Monitor Calls Abandoned Before Answered	
Event Notification	
Raise event if number of calls abandoned before answered exceeds threshold?	Select **Yes** to raise an event if the number of abandoned calls exceeds the threshold you set. The default is Yes. A call is considered abandoned when the caller hangs up before the call is answered.
Threshold - Maximum number of calls abandoned before answered	Specify the maximum number of calls that can be abandoned before an event is raised. The default is 10 calls.
Event severity when number of calls abandoned before answered exceeds threshold	Set the event severity level, from 1 to 40, to indicate the importance of an event in which the number of abandoned calls exceeds the threshold. The default is 10.
Data Collection	

Parameter	How to Set It
Collect data for number of calls abandoned before answered?	Select **Yes** to collect data for charts and reports. If enabled, data collection returns the number of calls that were abandoned before being answered by an attendant. The default is unselected.
Monitor Calls Abandoned While on Hold	
Event Notification	
Raise event if number of calls abandoned while on hold exceeds threshold?	Select **Yes** to raise an event if the number of calls that were abandoned while on hold exceeds the threshold you set. The default is Yes.
Threshold - Maximum number of calls abandoned while on hold	Specify the maximum number of on-hold calls that can be abandoned before an event is raised. The default is 10 calls.
Event severity when number of calls abandoned while on hold exceeds threshold	Set the event severity level, from 1 to 40, to indicate the importance of an event in which the number of calls that were abandoned while on hold exceeds the threshold. The default is 10.
Data Collection	
Collect data for number of calls abandoned before answered?	Select **Yes** to collect data for charts and reports. If enabled, data collection returns the number of calls that were abandoned before being answered by an attendant. The default is unselected.
Monitor Calls on Hold	
Event Notification	
Raise event if number of calls on hold exceeds threshold?	Select **Yes** to raise an event if the number of calls that were placed on hold exceeds the threshold you set. The default is Yes.
Threshold - Maximum number of calls on hold	Specify the maximum number of calls that can be placed on hold before an event is raised. The default is 100 calls.
Event severity when number of calls on hold exceeds threshold	Set the event severity level, from 1 to 40, to indicate the importance of an event in which the number of calls that were placed on hold exceeds the threshold. The default is 15.
Data Collection	
Collect data for number of calls on hold?	Select **Yes** to collect data for charts and reports. If enabled, data collection returns the number of calls that were placed on hold during the monitoring period. The default is Yes.
Monitor Queued Calls	
Event Notification	
Raise event if number of queued calls exceeds threshold?	Select **Yes** to raise an event if the number of calls in queue for an available attendant exceeds the threshold you set. The default is Yes.
Threshold - Maximum number of queued calls	Specify the maximum number of calls that can be in queue before an event is raised. The default is 100 calls.
Event severity when number of queued calls exceeds threshold	Set the event severity level, from 1 to 40, to indicate the importance of an event in which the number of calls in queue for an available attendant exceeds the threshold. The default is 15.

Parameter	How to Set It
Data Collection	
Collect data for number of queued calls?	Select **Yes** to collect data for charts and reports. If enabled, data collection returns the number of calls in queue during the monitoring period. The default is Yes.
Monitor Time Attendants are Active	
Event Notification	
Raise event if time attendants are active exceeds threshold?	Select **Yes** to raise an event if the number of minutes in which attendants are active (on a call) exceeds the threshold you set. The default is Yes.
Threshold - Maximum time attendants are active	Specify the maximum number of minutes attendants can be active before an event is raised. The default is 15 minutes.
Event severity when time attendants are active exceeds threshold	Set the event severity level, from 1 to 40, to indicate the importance of an event in which the number of minutes that attendants are active exceeds the threshold. The default is 15.
Data Collection	
Collect data for time attendants are active?	Select **Yes** to collect data for charts and reports. If enabled, data collection returns the number of minutes in which attendants were active during the monitoring period. The default is Yes.
Monitor Average Answer Time	
Event Notification	
Raise event if average answer time exceeds threshold?	Select **Yes** to raise an event if the average number of minutes that attendants take to answer calls exceeds the threshold you set. The default is Yes.
Threshold - Maximum average answer time	Specify the highest average number of minutes it can take attendants to answer calls before an event is raised. The default is 5 minutes.
Event severity when average answer time exceeds threshold	Select the event severity level, from 1 to 40, to indicate the importance of an event in which the average number of minutes it takes for attendants to answer calls exceeds the threshold. The default is 15.
Data Collection	
Collect data for average answer time?	Select **Yes** to collect data for charts and reports. If enabled, data collection returns the average number of minutes that attendants took to answer calls during the monitoring period. The default is Yes.

CallActivity

Use this Knowledge Script to monitor call activity on selected Communication Managers. This script raises an event if a threshold is exceeded. In addition, this script generates data streams for the number of active calls and the number of completed calls.

When you start the CallActivity Knowledge Script job, the managed object begins collecting call detail records (CDRs) to store in the Avaya CM supplemental database. After the CallActivity job stops, the managed object continues to collect CDRs. CDR collection stops within a time period equal to two intervals of the job, but never less than 4 minutes after the CallActivity job stops.

Prerequisite

Run SetupSupplementalDB to create the Avaya CM supplemental database.

Resource Object

AvayaCM Active SPE object

Default Schedule

By default, this script runs every 5 minutes.

Setting Parameter Values

Set the following parameters as needed:

Parameter	How to Set It
General Settings	
Job Failure Notification	
Event severity when job fails	Set the event severity level, from 1 to 40, to indicate the importance of the failure of the CallActivity job. The default is 5.
Monitor Active Calls	
Event Notification	
Raise event if number of active calls exceeds threshold?	Select **Yes** to raise an event if the number of active calls exceeds the threshold you set. The default is Yes.
Threshold -- Number of Active Calls	Specify the maximum number of calls that can be active before an event is raised. The default is 100 calls.
Event severity when number of active calls exceeds threshold	Set the event severity level, from 1 to 40, to indicate the importance of an event in which the number of active calls exceeds the threshold. The default is 15.
Data Collection	
Collect data for number of active calls?	Select **Yes** to collect data for charts and reports. If enabled, data collection returns the number of calls that were active during the monitoring period. The default is Yes.
Monitor Completed Calls	
Event Notification	
Raise event if number of completed calls exceeds threshold?	Select **Yes** to raise an event if the number of completed calls exceeds the threshold you set. The default is Yes.
Threshold - Number of Completed Calls	Specify the maximum number of calls that can be completed before an event is raised. The default is 100 calls.

Parameter	How to Set It
Event severity when number of completed calls exceeds threshold	Set the event severity level, from 1 to 40, to indicate the importance of an event in which the number of completed calls exceeds the threshold. The default is 15.
Data Collection	
Collect data for number of completed calls?	Select **Yes** to collect data for charts and reports. If enabled, data collection returns the number of calls that were completed during the monitoring period. The default is Yes.

CallFailures

Use this Knowledge Script to monitor call detail records (CDRs) in the Avaya CM supplemental database for calls that terminated with abnormal condition codes. You can indicate which condition codes should not be considered abnormal. This script raises an event if the number of failed calls exceeds the threshold or if the supplemental database contains no records. In addition, this script generates data streams for the number of failed calls.

When you start the CallFailures Knowledge Script job, the managed object begins collecting CDRs to store in the supplemental database. After the CallFailures job stops, the managed object continues to collect CDRs. CDR collection stops within a time period equal to two intervals of the job, but never less than 4 minutes after the CallFailures job stops.

The purpose of this script is twofold:

- **Monitoring**. In monitoring mode, this script checks the CDR tables at each specified interval for new records that match your query. In the first iteration of the job, this script finds the last record in the CDR table and checks back one interval from there. In subsequent iterations, this script checks for new records that match the query in each interval.
- **Troubleshooting**. In troubleshooting mode, this script runs once and checks the CDR tables for calls that disconnected within the range you select in the *Select call disconnect time range* parameter. Select **Run once** on the Schedule tab to run this script in troubleshooting mode. However, the managed object does not collect CDRs unless this script is running, which could pose a problem should you want to troubleshoot a call that occurred 5 minutes ago, for example. Therefore, to perform troubleshooting as needed, run this script as a separate job with data collection and events disabled, and set the schedule to run every *n* minutes. By doing so, the supplemental database will always contain data you can use for troubleshooting.

Prerequisite

Run SetupSupplementalDB to create the Avaya CM supplemental database.

Condition Codes

The following table identifies all supported condition codes:

Condition Code	Description
0	An intraswitch call, which originates and terminates on the switch
1 (A)	An attendant-handled call or an attendant-assisted call, except conference calls

Condition Code	Description
4 (D)	An extremely long call or a call with an extremely high message count TSC. An extremely long call is one that lasts for 10 or more hours. An extremely high message count TSC is 9999 or more messages. When a call exceeds 10 hours, the system creates a call record with this condition code and a duration of 9 hours, 59 minutes, and 1-9 tenths of a minute. The system creates a similar call record with this condition code after each succeeding 10-hour period. When the call terminates, the system creates a final call record with a different condition code that identifies the type of call.
6 (E)	A call the system did not record because system resources were unavailable. The CDR record includes the time and the duration of the outage. The system generates this condition code for: ◆ Calls that the system routes to the attendant ◆ Calls that require the CDR feature to overwrite records ◆ ISDN calls that are not completed at the far end, if the Q.931 message indicates the reason that the call was not completed. The system does not generate the condition code for ISDN calls that receive inband tones.
7 (G)	Calls that use the AAR or ARS feature.
8 (H)	Calls that are served on a delayed basis by the Ringback Queuing feature.
9 (I)	An incoming call, a tandem call, an incoming NCA-TSC call, or a tandem NCA-TSC call
A	An outgoing call
B	An adjunct-placed outgoing call
C (L)	A conference call For trunk CDR, the system create a separate call record, with this condition code, for each incoming or outgoing trunk that is used during the conference call. If you disable ITCS and OTCS, the system records the extension of the originator of the conference call. The system does not record any other extension. If you disable ITCS, and you administer the originator of the conference call to use Intraswitch CDR, the system generates a call with this condition code whenever the originator of the conference dials a nontrunk conference participant. If ITCS is active, and you do not administer the originator of the conference call to use Intraswitch CDR, the system generates a call with this condition code whenever the originator of the conference dials an intraswitch conference participant.

Condition Code	Description
E (N)	A call that the system does not complete because the following facilities to complete the call are unavailable: ◆ Outgoing calls - The trunks are busy and no queue exists. - The trunks are busy and the queue is full. ◆ Incoming calls - The extension is busy. - The extension is unassigned. This condition code also identifies an ISDN Call By Call Service Selection call that is unsuccessful because of an administered trunk usage allocation plan. Incoming trunk calls to a busy telephone do not generate a CDR record.
F	A call that the system does not complete because of one of the following conditions: ◆ The originator of the call has insufficient calling privileges. ◆ An NSF mismatch occurs for an ISDN call. ◆ An authorization mismatch occurs for a data call.
G	A call that the system terminates to a ringing station
H	Notes that the system abandoned a ring call
I	A call that the system terminates to a busy station
J	An incoming trunk call that is a new connection that uses Additional Network Feature-Path Replacement (ANF-PR) or DCS with Rerouting. For more information, see the *Administrator Guide for Avaya Communication Manager*.
K	An outgoing trunk call that is a new connection that uses ANF-PR or DCS with Rerouting. For more information, see the *Administrator Guide for Avaya Communication Manager*.
M	An outgoing trunk call that the system disconnects because the call exceeds the time allowed.
T	CDR records for calls that meet the following conditions: ◆ The **Condition Code 'T' for Redirected Calls?** field on the CDR System Parameters screen is set to **y**. ◆ The incoming trunk group is direct inward dialing (DID). ◆ The system automatically redirects an incoming call off of the server.

Resource Object

AvayaCM Active SPE object

Default Schedule

By default, this script runs every 5 minutes.

Setting Parameter Values

Set the following parameters as needed:

Parameter	How to Set It
General Settings	
Job Failure Notification	
Event severity when job fails	Set the event severity level, from 1 to 40, to indicate the importance of the failure of the CallFailures job. The default is 5.
Include call details?	Select **Yes** to include call details in the events raised by this script. Leave this parameter unselected to suppress the following call details: ◆ Condition Code ◆ Calling Number ◆ Called Number ◆ Connect Time ◆ Disconnect Time ◆ Duration **NOTE:** If you configured Communication Manager to use Agent ID numbers, an event will identify an Agent ID or gateway, rather than a called or calling phone extension. The default is selected.
Raise event if no records found?	Select **Yes** to raise an event if there are no CDRs to monitor. This does not mean there are no CDRs with abnormal condition codes. Instead, it means there are no CDRs at all. The default is unselected.
Event severity when no records found	Set the event severity level, from 1 to 40, to indicate the importance of an event in which no CDRs were found. The default is 25.
Query Filters	No matter how many calls match the filters you select, an event displays only the first 50 calls.
Exclude these failure codes	Provide a comma-separated list of the condition codes you do not want to monitor.
Exclude these failure codes on zero duration calls only	Provide a comma-separated list of the condition codes you do not want to monitor, but only for calls that have a duration of zero.
Include Only these FRL codes	Provide a comma-separated list of the Facilities Restriction Level (FRL) codes you want to use as filters. Leave this field blank to include all FRL codes. For more information, see Step 6 (Optional): Configuring CDR Parameters to Filter by FRL Codes. **NOTE:** To avoid an error message, run the SetupSupplementalDB Knowledge Script once before running this script. Also, set up FRL at the Avaya System Access Terminal (SAT) interface before running this script. Do not use FRL at the Avaya SAT interface with previous versions of this module.
Minimum duration	Set this parameter to filter out records whose call duration is less than the specified value. Accept the default of 0 to ignore the filter for minimum call duration.

Parameter	How to Set It
Maximum duration	Set this parameter to filter out records whose call duration is greater than or equal to the specified value. Accept the default of 0 to ignore the filter for maximum call duration.
Calling phone number	Specify the number of the calling phone you want to find in the CDRs. Wildcard characters are acceptable. Leave this parameter blank to search for any calling phone number. **NOTE:** If you configured Communication Manager to use Agent ID numbers, an event will identify the Agent ID or gateway associated with the *Calling phone number* you enter here.
Phone number connector	Set this parameter only if you specify both a *Calling phone number* and a *Called phone number*. Your selection indicates how the script will connect the two parameters: AND or OR. The default is AND.
Called phone number	Specify the number of the called phone you want to find in the CDRs. Wildcard characters are acceptable. Leave this parameter blank to search for any called phone number. **NOTE:** If you configured Communication Manager to use Agent ID numbers, an event will identify the Agent ID or gateway associated with the *Called phone number* you enter here.
Troubleshooting	
Call disconnect time range	Select a **Specific** or **Sliding** date/time range for which the query should search for data. The default time range is fixed at 24 hours. **NOTE:** This parameter is valid only when you select **Run once** on the Schedule tab.
Monitor Failed Calls	
Event Notification	
Raise event if number of failed calls exceeds threshold?	Select **Yes** to raise an event if the number of calls that failed with an abnormal condition code exceeds the threshold you set. The default is Yes.
Threshold - Maximum number of failed calls	Specify the maximum number of calls that can fail before an event is raised. The default is 0 calls.
Event severity when number of failed calls exceeds threshold	Set the event severity level, from 1 to 40, to indicate the importance of an event in which the number of failed calls exceeds the threshold. The default is 15.
Data Collection	
Collect data for number of failed calls?	Select **Yes** to collect data for charts and reports. If enabled, data collection returns the number of calls that failed with an abnormal condition code during the monitoring period. The default is unselected.

CallQuality

Use this Knowledge Script to monitor RTCP packets in the Avaya CM supplemental database for call quality statistics: jitter, latency, packet loss, MOS (Mean Opinion Score), and R-Value. This script raises an event if a monitored value exceeds or falls below a threshold. MOS and R-Value are computed only for calls that use one of the following codecs: G.711u, G.711a, or G.729.

NOTE: You can trigger NetIQ Vivinet Diagnostics to diagnose the problem indicated by an event in which the percentage lost data threshold is exceeded. For more information, see "Triggering Call and Phone Quality Diagnoses" on page 66.

When you start the CallQuality Knowledge Script job, the managed object begins collecting RTCP packets to store in the Avaya CM supplemental database. After the CallQuality job stops, the managed object continues to collect packets. Packet collection stops within a time period equal to two intervals of the job, but never less than 4 minutes after the CallQuality job stops.

The purpose of this script is twofold:

- **Monitoring**. In monitoring mode, this script checks the supplemental database tables at each specified interval for new records that match your query. In the first iteration of the job, this script finds the last record in a table and checks back one interval from there. In subsequent iterations, this script checks for new records that match the query in each interval.
- **Troubleshooting**. In troubleshooting mode, this script runs once and checks the supplemental database for calls that disconnected within the range you select in the *Select call disconnect time range* parameter. Select **Run once** on the Schedule tab to run this script in troubleshooting mode. However, the managed object does not collect RTCP packets unless this script is running, which could pose a problem should you want to troubleshoot a call that occurred 5 minutes ago, for example. Therefore, to perform troubleshooting as needed, run this script as a separate job with data collection and events disabled, and set the schedule to run every *n* minutes. By doing so, the Avaya CM supplemental database will always contain data you can use for troubleshooting.

Understanding data streams and Threshold Events

This script generates data streams for average MOS, R-Value, jitter, latency (one-way delay), and packet loss. These average values are based on data from each phone involved in calls that completed during the script's interval, which is, by default, every 5 minutes. For example, in a given call, the calling phone's jitter was 30 milliseconds and the called phone's jitter was 75 milliseconds. For this call, the data stream would be a calculated average of the jitter for both phones: 52.5 milliseconds.

This calculated average is below the default threshold value of 60 milliseconds. However, AppManager raises threshold events based on values for each phone in a call, not on the average value. Therefore, for this call, AppManager would raise one event based on the 75 milliseconds of jitter for the called phone.

Prerequisite

Run SetupSupplementalDB to create the Avaya CM supplemental database.

Resource Object

AvayaCM Active SPE object

Default Schedule

By default, this script runs every 5 minutes.

Setting Parameter Values

Set the following parameters as needed:

Parameter	How to Set It
General Settings	
Job Failure Notification	
Event severity when job fails	Set the event severity level, from 1 to 40, to indicate the importance of the failure of the CallQuality job. The default is 5.
Include call details?	Select **Yes** to include call details in the events raised by this script. Leave this parameter unselected to suppress the following call details: ◆ Caller and Called Average MOS ◆ Caller and Called Average R-Value ◆ Caller and Called Jitter ◆ Caller and Called Latency ◆ Caller and Called Lost Packets ◆ Caller and Called Codec ◆ Caller and Called Number ◆ Connect Time ◆ Disconnect Time ◆ Duration **NOTE:** If you configured Communication Manager to use Agent ID numbers, an event will identify an Agent ID or gateway, rather than a called or calling phone extension.
Raise event if no records found?	Select **Yes** to raise an event if there are no RTCP packets to monitor in the Avaya CM supplemental database. This does not mean there are no records with call quality data. It means there are no records at all. The default is unselected.
Event severity when no records found	Set the event severity level, from 1 to 40, to indicate the importance of an event in which no RTCP packets were found. The default is 25.
Query Filters	No matter how many calls match the filters you select, an event displays only the first 50 calls.
Minimum duration	Use this parameter to filter out records whose call duration is less than the value you specify. Accept the default of 0 seconds to ignore the filter for minimum duration.
Maximum table size	Specify the maximum number of rows of results to include in the call quality tables shown in the details of an event message. The default is 25 rows.
Maximum duration	Use this parameter to filter out records whose call duration is greater than or equal to the value you specify. Accept the default of 0 seconds to ignore the filter for maximum duration.

Parameter	How to Set It
Calling phone number	Specify the number of the calling phone you want to find in the supplemental database. Wildcard characters are acceptable. Leave this parameter blank to search for any calling phone number. **NOTE:** If you configured Communication Manager to use Agent ID numbers, an event will identify the Agent ID or gateway associated with the *Calling phone number* you enter here.
Phone number connector	Set this parameter only if you specify both a *Calling phone number* and a *Called phone number*. Your selection indicates how the script will connect the two parameters: AND or OR. The default is AND.
Called phone number	Specify the number of the called phone you want to find in the supplemental database. Wildcard characters are acceptable. Leave this parameter blank to search for any called phone number. **NOTE:** If you configured Communication Manager to use Agent ID numbers, an event will identify the Agent ID or gateway associated with the *Called phone number* you enter here.
Troubleshooting	
Call disconnect time range	Select a range of time and dates in which the query should search for data in the supplemental database. ◆ Select **Fixed Time** to select specific days and times that the query should begin and end. ◆ Select **Sliding** to select a number of hours, days, months, or years in which to search. The query starts its search at the time the job runs, and goes back through the supplemental database for the number of units you select. The default is Fixed Time. **NOTE:** This parameter is valid only when you select **Run once** on the Schedule tab.
Monitor Average MOS	
Event Notification	
Raise event if average MOS falls below threshold?	Select **Yes** to raise an event if the average MOS value falls below the threshold. The default is Yes.
Threshold - Average MOS	Specify the lowest average MOS value that must occur to prevent an event from being raised. The default is 3.60.
Event severity when average MOS falls below threshold	Set the event severity level, from 1 to 40, to indicate the importance of an event in which the average MOS value falls below the threshold. The default is 5.
Minimum number of calls with a MOS value below the threshold which must be found for an event to be raised for the interval	Set the minimum number of calls, from 1 to 1000, which must be determined to have a MOS value below the threshold, to raise an event for the interval. The default is 1.
Data Collection	
Collect data for average MOS?	Select **Yes** to collect data for charts and reports. If enabled, data collection returns the average MOS value during the monitoring period. The default is unselected.

Parameter	How to Set It
Monitor Average R-Value	
Event Notification	
Raise event if average R-Value falls below threshold?	Select **Yes** to raise an event if the average R-Value falls below the threshold. The default is Yes.
Threshold - Average R-Value	Specify the lowest average R-Value that must occur to prevent an event from being raised. The default is 70.
Event severity when average R-Value falls below threshold	Set the event severity level, from 1 to 40, to indicate the importance of an event in which the average R-Value falls below the threshold. The default is 5.
Minimum number of calls with an RValue value below the threshold which must be found for an event to be raised for the interval	Set the minimum number of calls, from 1 to 1000, which must be determined to have an RValue value below the threshold, to raise an event for the interval. The default is 1.
Data Collection	
Collect data for average R-Value?	Select **Yes** to collect data for charts and reports. If enabled, data collection returns the average R-Value during the monitoring period. The default is unselected.
Monitor Average Jitter	
Event Notification	
Raise event if jitter exceeds threshold?	Select **Yes** to raise an event if the jitter value exceeds the threshold. The default is Yes.
Threshold - Maximum jitter	Specify the highest average jitter value that can occur before an event is raised. The default is 60 milliseconds.
Event severity when jitter exceeds threshold	Set the event severity level, from 1 to 40, to indicate the importance of an event in which the jitter value exceeds the threshold. The default is 15.
Minimum number of calls with a latency value exceeding the threshold which must be found for an event to be raised for the interval	Set the minimum number of calls, from 1 to 1000, which must be determined to have a latency value exceeding the threshold, to raise an event for the interval. The default is 1.
Data Collection	
Collect data for jitter?	Select **Yes** to collect data for charts and reports. If enabled, data collection returns the amount of jitter that occurred during the monitoring period. The default is unselected.
Monitor Average Latency	
Event Notification	
Raise event if latency exceeds threshold?	Select **Yes** to raise an event if the latency value exceeds the threshold. The default is Yes.
Threshold - Maximum latency	Specify the highest amount of average latency that can occur before an event is raised. The default is 400 milliseconds.

Parameter	How to Set It
Event severity when latency exceeds threshold	Set the event severity level, from 1 to 40, to indicate the importance of an event in which the latency value exceeds the threshold. The default is 15.
Minimum number of calls with a latency value exceeding the threshold which must be found for an event to be raised for the interval	Set the minimum number of calls, from 1 to 1000, which must be determined to have a latency value exceeding the threshold, to raise an event for the interval. The default is 1.
Data Collection	
Collect data for latency?	Select **Yes** to collect data for charts and reports. If enabled, data collection returns the amount of latency that occurred during the monitoring period. The default is unselected.
Monitor Average Packet Loss	
Event Notification	
Raise event if packet loss exceeds threshold?	Select **Yes** to raise an event if the packet loss value exceeds the threshold. The default is Yes.
Threshold - Maximum packet loss	Specify the highest percentage of average packet loss that can occur before an event is raised. The default is 1%.
Event severity when packet loss exceeds threshold	Set the event severity level, from 1 to 40, to indicate the importance of an event in which the packet loss value exceeds the threshold. The default is 15.
Minimum number of calls with a packet loss value exceeding the threshold which must be found for an event to be raised for the interval	Set the minimum number of calls, from 1 to 1000, which must be determined to have a packet loss value exceeding the threshold, to raise an event for the interval. The default is 1.
Data Collection	
Collect data for packet loss?	Select **Yes** to collect data for charts and reports. If enabled, data collection returns the percentage of packet loss that occurred during the monitoring period. The default is unselected.

Triggering Call and Phone Quality Diagnoses

You can use NetIQ Vivinet Diagnostics to diagnose problems identified by AvayaCM Knowledge Scripts.

Using the existing methodology of launching an Action script based on an event, AppManager can launch Action_DiagnoseVoIPQuality to trigger Vivinet Diagnostics to diagnose the problem for events raised by the following Knowledge Scripts:

- **AvayaCM_CallQuality** events trigger Vivinet Diagnostics to diagnose the problem when average MOS, average R-Value, average jitter, average latency, and average packet loss fall below or exceed their thresholds.
- **AvayaCM_PhoneQuality** events trigger Vivinet Diagnostics to diagnose the problem when MOS, R-Value, jitter, latency, and packet loss fall below or exceed their thresholds during the data collection interval.

The Action script runs by default only if Vivinet Diagnostics 2.3 or later is installed on the computer on which the script is running.

To trigger Vivinet Diagnostics:

1. When setting parameter values for the PhoneQuality or CallQuality scripts, click the **Actions** tab. Action_DiagnoseVoIPQuality is selected by default.
2. Click **Properties** and enter values for all parameters for the Action script. For more information about the parameter values, click **Help** on the Properties for Action_DiagnoseVoIPQuality dialog box.

For more information about Vivinet Diagnostics and call quality diagnoses, see the *User Guide for Vivinet Diagnostics* and the Help for the Action_DiagnoseVoIPQuality Knowledge Script.

CallQuery

Use this Knowledge Script to search for call detail records (CDRs) retrieved from Communication Manager and stored in the Avaya CM supplemental database. The search is based on query filters you select. This script raises an event if no CDRs are found or if the number of CDRs found exceeds the threshold you set. In addition, this script generates a data stream for the number of records found.

When you start the CallQuery Knowledge Script job, the managed object begins collecting CDRs to store in the Avaya CM supplemental database. After the CallQuery job stops, the managed object continues to collect CDRs. CDR collection stops within a time period equal to two intervals of the job, but never less than 4 minutes after the CallQuery job stops.

The purpose of this script is twofold:

- **Monitoring**. In monitoring mode, this script checks the CDR tables at each specified interval for new records that match your query. In the first iteration of the job, this script finds the last record in the CDR table and checks back one interval from there. In subsequent iterations, this script checks for new records that match the query in each interval.
- **Troubleshooting**. In troubleshooting mode, this script runs once and checks the CDR tables for calls that disconnected within the range you select in the *Select call disconnect time range* parameter. Select **Run once** on the Schedule tab to run this script in troubleshooting mode. However, the managed object does not collect CDRs unless this script is running, which could pose a problem should you want to troubleshoot a call that occurred 5 minutes ago, for example. Therefore, to perform troubleshooting as needed, run this script as a separate job with data collection and events disabled, and set the schedule to run every *n* minutes. By doing so, the Avaya CM supplemental database will always contain data you can use for troubleshooting.

Prerequisite

Run SetupSupplementalDB to create the Avaya CM supplemental database.

Resource Object

AvayaCM Active SPE object

Default Schedule

By default, this script runs every 5 minutes.

Setting Parameter Values

Set the following parameters as needed:

Parameter	How to Set It
General Settings	
Job Failure Notification	
Event severity when job fails	Set the event severity level, from 1 to 40, to indicate the importance of the failure of the CallQuery job. The default is 5.
Include call details?	Select **Yes** to include call details in the events raised by this script. Leave this parameter unselected to suppress the following call details: ◆ Condition Code ◆ Calling Number ◆ Called Number ◆ Connect Time ◆ Disconnect Time ◆ Duration (seconds)
Raise event if no records found?	Select **Yes** to raise an event if there are no CDRs to monitor. This does not mean there are no CDRs with call quality data. It means there are no CDRs at all. The default is unselected.
Event severity when no records found	Set the event severity level, from 1 to 40, to indicate the importance of an event in which no CDRs were found. The default is 25.
Query Filters	No matter how many calls match the filters you select, an event displays only the first 50 calls.
Maximum table size	Specify the maximum number of detail rows to include in an event message. The default is 50 rows.
Minimum duration	Set this parameter to filter out records whose call duration is less than the specified value. Accept the default of 0 to ignore the filter for minimum call duration.
Maximum duration	Set this parameter to filter out records whose call duration is greater than or equal to the specified value. Accept the default of 0 to ignore the filter for maximum call duration.
Calling phone number	Specify the number of the calling phone you want to find in the CDRs. Wildcard characters are acceptable. Leave this parameter blank to search for any calling phone number. **NOTE:** If you configured Communication Manager to use Agent ID numbers, an event will identify the Agent ID or gateway associated with the *Calling phone number* you enter here.
Phone number connector	Set this parameter only if you specify both a *Calling phone number* and a *Called phone number*. Your selection indicates how the script will connect the two parameters: AND or OR. The default is AND.

Parameter	How to Set It
Called phone number	Specify the number of the called phone you want to find in the CDRs. Wildcard characters are acceptable. Leave this parameter blank to search for any called phone number. **NOTE:** If you configured Communication Manager to use Agent ID numbers, an event will identify the Agent ID or gateway associated with the *Called phone number* you enter here.
Troubleshooting	
Call disconnect time range	Select a **Specific** or **Sliding** date/time range for which the query should search for data. The default time range is fixed at 24 hours. **NOTE:** This parameter is valid only when you select **Run once** on the Schedule tab.
Monitor Records Found	
Event Notification	
Raise event if number of records exceeds threshold?	Select **Yes** to raise an event if the number of CDRs found exceeds the threshold. The default is Yes.
Threshold - Maximum number of records	Specify the maximum number of CDRs that can be found before an event is raised. The default is 0 CDRs.
Event severity when number of records exceeds threshold	Set the event severity level, from 1 to 40, to indicate the importance of an event in which the number of CDRs found exceeds the threshold. The default is 15.
Data Collection	
Collect data for number of records?	Select **Yes** to collect data for charts and reports. If enabled, data collection returns the number of CDRs found during the monitoring period.

CPU_Usage

Use this Knowledge Script to monitor a Communication Manager server for system management CPU usage, operating system CPU usage, call-processing CPU usage, and available CPU. Note that “available CPU” is the amount of CPU that is available for high-priority tasks, including CPU allocated for low-priority tasks. “Available CPU” is not the amount of CPU that is left over after system management, operating system, and call processing take their shares. Therefore, the four CPU usage values provided by this script can total more than 100%.

This script raises events when values exceed or fall below thresholds that you set. In addition, this script generates data streams for all monitored metrics.

NOTE: If you bypass SNMP to discover a call manager (by selecting the **Discover using manual configuration?** parameter), this script is not available.

Resource Object

AvayaCM Active SPE object

Default Schedule

By default, this script runs every 2 minutes.

Setting Parameter Values

Set the following parameters as needed:

Parameter	How to Set It
General Settings	
Job Failure Notification	
Event severity when job fails	Set the event severity level, from 1 to 40, to indicate the importance of the failure of the CPU_Usage job. The default is 5.
Monitor Call Processing CPU Usage	
Event Notification	
Raise event if call processing CPU usage exceeds threshold?	Select **Yes** to raise an event if the percentage of CPU used by call processing exceeds the threshold you set. The default is Yes.
Threshold - Maximum call processing CPU usage	Specify the maximum amount of CPU that call processing can use before an event is raised. The default is 80%.
Event severity when call processing CPU usage exceeds threshold	Set the event severity level, from 1 to 40, to indicate the importance of an event in which the percentage of CPU used by call processing exceeds the threshold. The default is 15.
Data Collection	
Collect data for call processing CPU usage?	Select **Yes** to collect data for charts and reports. If enabled, data collection returns the percentage of CPU used by call processing for the monitoring period. The default is Yes.
Monitor System Management CPU Usage	
Event Notification	
Raise event if system management CPU usage exceeds threshold?	Select **Yes** to raise an event if the percentage of CPU used by system management processes exceeds the threshold you set. The default is Yes.
Threshold - Maximum system management CPU usage	Specify the maximum percentage of CPU that system management processes can use before an event is raised. The default is 20%.
Event severity when system management CPU usage exceeds threshold	Set the event severity level, from 1 to 40, to indicate the importance of an event in which the percentage of CPU used by system management processes exceeds the threshold. The default is 15.
Data Collection	
Collect data for system management CPU usage?	Select **Yes** to collect data for charts and reports. If enabled, data collection returns the percentage of CPU used by system management processes for the monitoring period. The default is unselected.
Monitor Operating System CPU Usage	
Event Notification	

Parameter	How to Set It
Raise event if operating system CPU usage exceeds threshold?	Select **Yes** to raise an event if the percentage of CPU used by operating system processes exceeds the threshold you set. The default is Yes.
Threshold - Maximum operating system CPU usage	Specify the maximum percentage of CPU that operating system processes can use before an event is raised. The default is 20%.
Event severity when operating system CPU usage exceeds threshold	Set the event severity level, from 1 to 40, to indicate the importance of an event in which the percentage of CPU used by operating system processes exceeds the threshold. The default is 15.
Data Collection	
Collect data for operating system CPU usage?	Select **Yes** to collect data for charts and reports. If enabled, data collection returns the percentage of CPU used by operating system processes for the monitoring period. The default is unselected.
Monitor Available CPU	
Event Notification	
Raise event if available CPU falls below threshold?	Select **Yes** to raise an event if the percentage of CPU that is available for Communication Manager falls below the threshold you set. The default is Yes.
Threshold - Minimum available CPU	Specify the minimum amount of CPU that must be available for Communication Manager before an event is raised. The default is 20%.
Event severity when available CPU falls below threshold	Set the event severity level, from 1 to 40, to indicate the importance of an event in which available CPU falls below the threshold. The default is 15.
Data Collection	
Collect data for available CPU?	Select **Yes** to collect data for charts and reports. If enabled, data collection returns the percentage of available CPU for the monitoring period. The default is unselected.

DeleteCDRRecords

Use this Knowledge Script to delete call detail records (CDR) from supplemental database. This script raises an event if any failure is encountered.

Before running this KS, make sure to run Discovery (with setup database option) or SetupSupplementalDB KS to setup supplemental database first. When you run the DeleteCDRrecords Knowledge Script job, it starts deleting call detail records from supplemental database that are older than n days. Here, n is equal to "Number of days to keep call detail records" parameter you set while running SetupSupplementalDB KS.

Prerequisite

Run Discovery (with setup database option) or SetupSupplementalDB to create the Avaya CM supplemental database.

Resource Object

AvayaCM Active SPE object

Default Schedule

By default, this script runs every 5 minutes

Setting Parameter Values

Set the following parameter as needed:

Parameter	How to Set It
General Settings	
Job Failure Notification	
Event severity when job fails	Set the event severity level, from 1 to 40, to indicate the importance of the failure of the DeleteCDRecords job. The default is 5.

ESS_Status

Use this Knowledge Script to monitor the registration status of an Avaya Enterprise Survivable Server (ESS). An ESS allows a media server to be used as a backup controller to protect Communication Manager against catastrophic failure.

This script raises an event if the server deregisters or reregisters. In addition, this script generates a data point for server registration status: 0 for deregistered and 1 for reregistered.

NOTE: If you bypass SNMP to discover a call manager (by selecting the **Discover using manual configuration?** parameter), this script is not available.

Resource Object

AvayaCM ESS object

Default Schedule

By default, this script runs every 5 minutes.

Setting Parameter Values

Set the following parameters as needed:

Parameter	How to Set It
General Settings	

Parameter	How to Set It
Job Failure Notification	
Event severity when job fails	Set the event severity level, from 1 to 40, to indicate the importance of the failure of the ESS_Status job. The default is 5.
Enable use of SNMP GETBulk operations?	Select **Yes** to allow this Knowledge Script to access Communication Manager MIBs using `GETNext` and `GETBulk` SNMP requests, as appropriate. The default is Yes. Deselect the check box to allow the Knowledge Script to use only `GETNext` requests. Not all MIB tables are extensive enough to need a `GETBulk` request. A `GETBulk` request is faster, but more CPU-intensive than a `GETNext` request.
Number of rows to request for each GETBulk operation	Specify the number of rows from the MIB table to return in a `GETBulk` request. The default is 10 rows. The number of rows determines how fast the MIB data is returned. If the CPU usage is too high, you can reduce the number of rows per `GETBulk` request or deselect the *Enable use of SNMP GETBulk requests?* parameter.
Interval to pause between GETBulk operations	Specify the number of milliseconds to wait between GETBulk requests. The default is 100 milliseconds. The length of the delay can help manage CPU usage and the speed of SNMP requests. For example, a one-row `GETBUlk` with a 100-millisecond delay between requests executes more slowly and uses less CPU than a `GETNext` request.
Monitor Registration Status	
Data Collection	
Collect data for registration status?	Select **Yes** to collect data for charts and reports. If enabled, data collection returns a 0 if the ESS deregisters and a 1 if the ESS reregisters. The default is unselected.
Event Notification	
Raise event if server deregisters?	Select **Yes** to raise an event if the ESS deregisters from Communication Manager. The default is Yes.
Event severity when server deregisters	Set the event severity level, from 1 to 40, to indicate the importance of an event in which the ESS deregisters from Communication Manager. The default is 15.
Raise event if server reregisters?	Select **Yes** to raise an event if the ESS reregisters with Communication Manager. The default is Yes.
Event severity when server reregisters	Set the event severity level, from 1 to 40, to indicate the importance of an event in which the ESS reregisters with Communication Manager. The default is 25.

H248GatewayStatus

Use this Knowledge Script to monitor the status of H.248 media gateways, including the following items:

- Major alarms

- Minor alarms
- Warning alarms
- Status of H.248 (server to gateway) link

This script raises an event if a threshold is exceeded. In addition, this script generates data streams for each monitored value.

NOTE: If you bypass SNMP to discover a call manager (by selecting the **Discover using manual configuration?** parameter), this script is not available.

Resource Object

AvayaCM H.248 Media Gateway object

Default Schedule

By default, this script runs every 5 minutes.

Setting Parameter Values

Set the following parameters as needed:

Parameter	How to Set It
General Settings	
Job Failure Notification	
Event severity when job fails	Set the event severity level, from 1 to 40, to indicate the importance of the failure of the H248GatewayStatus job. The default is 5.
Enable use of SNMP GETBulk operations?	Select **Yes** to allow this Knowledge Script to access Communication Manager MIBs using `GETNext` and `GETBulk` SNMP requests, as appropriate. The default is Yes. Deselect the check box to allow the Knowledge Script to use only `GETNext` requests. Not all MIB tables are extensive enough to need a `GETBulk` request. A `GETBulk` request is faster, but more CPU-intensive than a `GETNext` request.
Number of rows to request for each GETBulk operation	Specify the number of rows from the MIB table to return in a `GETBulk` request. The default is 10 rows. The number of rows determines how fast the MIB data is returned. If the CPU usage is too high, you can reduce the number of rows per `GETBulk` request or deselect the *Enable use of SNMP GETBulk requests?* parameter.
Interval to pause between GETBulk operations	Specify the number of milliseconds to wait between GETBulk requests. The default is 100 milliseconds. The length of the delay can help manage CPU usage and the speed of SNMP requests. For example, a one-row `GETBUlk` with a 100-millisecond delay between requests executes more slowly and uses less CPU than a `GETNext` request.

Parameter	How to Set It
Monitor Major Alarms	
Event Notification	
Raise event if number of major alarms exceeds threshold?	Select **Yes** to raise an event if the number of major alarms exceeds the threshold you set. The default is Yes.
Threshold - Maximum major alarms	Specify the highest number of major alarms that can occur before an event is raised. The default is 1 alarm.
Event severity when number of major alarms exceeds threshold	Set the event severity level, from 1 to 40, to indicate the importance of an event in which the number of major alarms exceeds the threshold. The default is 5.
Data Collection	
Collect data for major alarms?	Select **Yes** to collect data for charts and reports. If enabled, data collection returns the number of major alarms that occurred during the monitoring period. The default is Yes.
Monitor Minor Alarms	
Event Notification	
Raise event if number of minor alarms exceeds threshold?	Select **Yes** to raise an event if the number of minor alarms exceeds the threshold you set. The default is Yes.
Threshold - Maximum minor alarms	Specify the highest number of minor alarms that can occur before an event is raised. The default is 5 alarms.
Event severity when number of minor alarms exceeds threshold	Set the event severity level, from 1 to 40, to indicate the importance of an event in which the number of minor alarms exceeds the threshold. The default is 10.
Data Collection	
Collect data for minor alarms?	Select **Yes** to collect data for charts and reports. If enabled, data collection returns the number of minor alarms that occurred during the monitoring period. The default is Yes.
Monitor Warning Alarms	
Event Notification	
Raise event if number of warning alarms exceeds threshold?	Select **Yes** to raise an event if the number of warning alarms exceeds the threshold you set. The default is Yes.
Threshold - Maximum warning alarms	Specify the highest number of warning alarms that can occur before an event is raised. The default is 10 alarms.
Event severity when number of warning alarms exceeds threshold	Set the event severity level, from 1 to 40, to indicate the importance of an event in which the number of warning alarms exceeds the threshold. The default is 15.
Data Collection	
Collect data for warning alarms?	Select **Yes** to collect data for charts and reports. If enabled, data collection returns the number of warning alarms that occurred during the monitoring period. The default is unselected.

Parameter	How to Set It
Monitor H.248 Link Status	
Event Notification	
Raise event if H.248 link is down?	Select **Yes** to raise an event if the H.248 link is unavailable. The default is Yes.
Event severity when H.248 link is down	Set the event severity level, from 1 to 40, to indicate the importance of an event in which the H.248 link is unavailable.
Data Collection	
Collect data for link status?	Select **Yes** to collect data for charts and reports. If enabled, data collection returns 0 if the link is up or 1 if the link is down. The default is Yes.

HuntGroupUsage

Use this Knowledge Script to monitor the status of hunt groups. This script raises an event if a monitored value exceeds the threshold you set. In addition, this script generates data streams for the following statistics:

- Number of answered calls
- Number of queued calls
- Number of abandoned calls
- Average call wait time

NOTE: If you bypass SNMP to discover a call manager (by selecting the **Discover using manual configuration?** parameter), this script is not available.

Resource Object

AvayaCM Hunt Group object

Default Schedule

By default, this script runs every hour because the SNMP data it monitors is updated only once an hour. If you change the schedule to a shorter interval, you may receive SNMP request errors until the SNMP data is repopulated.

Setting Parameter Values

Set the following parameters as needed:

Parameter	How to Set It
General Settings	
Job Failure Notification	

Parameter	How to Set It
Event severity when job fails	Set the event severity level, from 1 to 40, to indicate the importance of the failure of the HuntGroupUsage job. The default is 5.
Enable use of SNMP GETBulk operations?	Select **Yes** to allow this Knowledge Script to access Communication Manager MIBs using `GETNext` and `GETBulk` SNMP requests, as appropriate. The default is Yes. Deselect the check box to allow the Knowledge Script to use only `GETNext` requests. Not all MIB tables are extensive enough to need a `GETBulk` request. A `GETBulk` request is faster, but more CPU-intensive than a `GETNext` request.
Number of rows to request for each GETBulk operation	Specify the number of rows from the MIB table to return in a `GETBulk` request. The default is 10 rows. The number of rows determines how fast the MIB data is returned. If the CPU usage is too high, you can reduce the number of rows per `GETBulk` request or deselect the *Enable use of SNMP GETBulk requests?* parameter.
Interval to pause between GETBulk operations	Specify the number of milliseconds to wait between GETBulk requests. The default is 100 milliseconds. The length of the delay can help manage CPU usage and the speed of SNMP requests. For example, a one-row `GETBUlk` with a 100-millisecond delay between requests executes more slowly and uses less CPU than a `GETNext` request.
Monitor Answered Calls	
Event Notification	
Raise event if number of answered calls exceeds threshold?	Select **Yes** to raise an event if the number of calls answered in the hunt group exceeds the threshold you set. The default is Yes.
Threshold - Maximum number of answered calls	Specify the highest number of calls that can be answered in the hunt group before an event is raised. The default is 100 calls.
Event severity when number of answered calls exceeds threshold	Set the event severity level, from 1 to 40, to indicate the importance of an event in which the number of calls answered in the hunt group exceeds the threshold. The default is 15.
Data Collection	
Collect data for number of answered calls?	Select **Yes** to collect data for charts and reports. If enabled, data collection returns the number of calls answered in the hunt group during the monitoring period. The default is Yes.
Monitor Abandoned Calls	
Event Notification	
Raise event if number of abandoned calls exceeds threshold?	Select **Yes** to raise an event if the number of calls abandoned in the hunt group exceeds the threshold you set. The default is Yes. A call is considered abandoned when the caller hangs up before the call is answered.
Threshold - Maximum number of abandoned calls	Specify the highest number of calls that can be abandoned before an event is raised. The default is 5 calls.

Parameter	How to Set It
Event severity when number of abandoned calls exceeds threshold	Set the event severity level, from 1 to 40, to indicate the importance of an event in which the number of abandoned calls exceeds the threshold. The default is 15.
Data Collection	
Collect data for number of abandoned calls?	Select **Yes** to collect data for charts and reports. If enabled, data collection returns the number of calls that were abandoned during the monitoring period. The default is unselected.
Monitor Queued Calls	
Event Notification	
Raise event if number of queued calls exceeds threshold?	Select **Yes** to raise an event if the number of calls in queue exceeds the threshold you set. The default is Yes.
Threshold - Maximum number of queued calls	Specify the highest number of calls that can be in queue before an event is raised. The default is 100 calls.
Event severity when number of queued calls exceeds threshold	Set the event severity level, from 1 to 40, to indicate the importance of an event in which the number of calls in queue exceeds the threshold. The default is 15.
Data Collection	
Collect data for number queued calls?	Select **Yes** to collect data for charts and reports. If enabled, data collection returns the number of calls that were in queue during the monitoring period. The default is Yes.
Monitor Average Call Wait Time	
Event Notification	
Raise event if average call wait time exceeds threshold?	Select **Yes** to raise an event if the average amount of call wait time exceeds the threshold you set. The default is Yes. Call wait time is the amount of time a call waits before being answered.
Threshold - Maximum average call wait time	Specify the highest amount of average wait time that calls can experience before an event is raised. The default is 1 minute.
Event severity when average call wait time exceeds threshold	Set the event severity level, from 1 to 40, to indicate the importance of an event in which the average amount of call wait time exceeds the threshold. The default is 15.
Data Collection	
Collect data for average call wait time?	Select **Yes** to collect data for charts and reports. If enabled, data collection returns the average amount of wait time that calls experienced during the monitoring period. The default is unselected.

LSP_Status

Use this Knowledge Script to monitor the registration status of an Avaya Local Survivable Processor (LSP). An LSP allows a media server to be a survivable call-processing server for remote and branch customer locations.

This script raises an event if the processor deregisters or reregisters. In addition, this script generates a data point for processor registration status: 0 for deregistered and 1 for reregistered.

NOTE: If you bypass SNMP to discover a call manager (by selecting the **Discover using manual configuration?** parameter), this script is not available.

Resource Object

AvayaCM LSP object

Default Schedule

By default, this script runs every 5 minutes.

Setting Parameter Values

Set the following parameters as needed:

Parameter	How to Set It
General Settings	
Job Failure Notification	
Event severity when job fails	Set the event severity level, from 1 to 40, to indicate the importance of the failure of the LSP_Status job. The default is 5.
Enable use of SNMP GETBulk operations?	Select **Yes** to allow this Knowledge Script to access Communication Manager MIBs using `GETNext` and `GETBulk` SNMP requests, as appropriate. The default is Yes. Deselect the check box to allow the Knowledge Script to use only `GETNext` requests. Not all MIB tables are extensive enough to need a `GETBulk` request. A `GETBulk` request is faster, but more CPU-intensive than a `GETNext` request.
Number of rows to request for each GETBulk operation	Specify the number of rows from the MIB table to return in a `GETBulk` request. The default is 10 rows. The number of rows determines how fast the MIB data is returned. If the CPU usage is too high, you can reduce the number of rows per `GETBulk` request or deselect the *Enable use of SNMP GETBulk requests?* parameter.

Parameter	How to Set It
Interval to pause between GETBulk operations	Specify the number of milliseconds to wait between GETBulk requests. The default is 100 milliseconds. The length of the delay can help manage CPU usage and the speed of SNMP requests. For example, a one-row `GETBUlk` with a 100-millisecond delay between requests executes more slowly and uses less CPU than a `GETNext` request.
Monitor Registration Status	
Data Collection	
Collect data for registration status?	Select **Yes** to collect data for charts and reports. If enabled, data collection returns a 0 if the LSP deregisters and a 1 if the LSP reregisters. The default is unselected.
Event Notification	
Raise event if processor deregisters?	Select **Yes** to raise an event if the LSP deregisters from Communication Manager. The default is Yes.
Event severity when processor deregisters	Set the event severity level, from 1 to 40, to indicate the importance of an event in which the LSP deregisters from Communication Manager. The default is 15.
Raise event if processor reregisters?	Select **Yes** to raise an event if the LSP reregisters with Communication Manager The default is Yes.
Event severity when processor reregisters	Set the event severity level, from 1 to 40, to indicate the importance of an event in which the LSP reregisters with Communication Manager. The default is 25.

PhoneConnectivity

Use this Knowledge Script to monitor disconnected registered phones for a Communication Manager and to maintain a history of the monitored phones in the Avaya CM supplemental database. This script raises an event if the number or percentage of disconnected registered phones exceeds the threshold you set. You can group events by cluster, building, floor, or type of phone.

NOTE: If you bypass SNMP to discover a call manager, this script is not available.

Prerequisites

- Run SetupSupplementalDB to create the Avaya CM supplemental database.
- Run RetrieveConfigData to populate the Avaya CM supplemental database.

Resource Object

AvayaCM_ActiveSPE

Default Schedule

By default, this script runs every 10 minutes.

Setting Parameter Values

Set the following parameters as needed:

Parameter	How to Set It
General Settings	
Job Failure Notification	
Event severity when job fails	Set the event severity level, from 1 to 40, to indicate the importance of the failure of the PhoneConnectivity job. The default is 5.
Enable use of SNMP GETBulk operations?	By default, this parameter is enabled, allowing the PhoneConnectivity Knowledge Script job to access Communication Manager MIBs using `GETNext` and `GETBulk` SNMP requests, as appropriate. Disable this parameter to allow the script to use only `GETNext` requests. Not all MIB tables are extensive enough to need a `GETBulk` request. A `GETBulk` request is faster, but more CPU-intensive than `GETNext`.
Number of rows to request for each GETBulk operation	Specify the number of rows from the MIB table to return in a `GETBulk` request. The default is 10 rows. The number of rows determines how much faster MIB data is returned. If CPU usage is too high, you can reduce the number of rows per `GETBulk` or disable the *Enable use of SNMP GETBulk requests?* parameter.
Interval to pause between GETBulk operations	Specify the number of milliseconds to wait between GETBulk requests. The default is 100 milliseconds. The length of delay can help manage CPU usage and speed of SNMP requests. For example, a one-row `GETBulk` with a 100-millisecond delay between requests runs more slowly and uses less CPU than a `GETNext` request.
Event Notification	
Raise event if disconnected registered phones in group exceed threshold?	Select **Yes** to raise an event if the number or percentage of disconnected registered phones in a group exceeds the threshold you set. The default is Yes. Use *Type of threshold* to create a “number” or “percentage” event. Use *Select event grouping* to select how to group the deregistered phones.

Parameter	How to Set It
Select event grouping	Select whether to group disconnected registered phones by **Cluster**, **Building**, **Floor,** or **Type** of phone. AppManager raises an event when the number or percentage of phones in *each* group exceeds the threshold you set. For example, suppose the following were true: ◆ You set *Maximum number of disconnected registered phones in the group* to "5" ◆ You set *Select event grouping* to "Building," and you have three buildings. If AppManager detects six disconnected registered phones in the first building, two in the second, and seven in the third, it will raise two events: one for the six disconnected registered phones in the first building and another for the seven disconnected registered phones in the third building. Because you set the threshold to "5," no event is raised for the disconnected registered phones in the second building. The default is Cluster.
Type of threshold	Select whether you want to raise events based on the **Number** or **Percent** of disconnected registered phones. The default is Number.
Threshold - Maximum number of disconnected registered phones	Use this parameter if you selected **Number** in *Type of threshold*. Specify the maximum number of registered phones that can be disconnected before an event is raised. The default is 0 phones.
Threshold - Maximum percent of disconnected registered phones	Use this parameter if you selected **Percent** in *Type of threshold*. Specify the maximum percentage of registered phones that can be disconnected before an event is raised. The default is 0%.
Event severity when disconnected registered phones exceed threshold	Set the event severity, from 1 to 40, to indicate the importance of an event in which the number or percentage of disconnected registered phones in a group exceeds the threshold you set. The default is 15.
Include phone details in event message?	Select **Yes** to include details of the disconnected registered phones in the event message. Phone details can include station extension, station name, building, floor, station type, room, cable, jack, port, and deregistration time. The default is Yes.
Maximum number of detail rows to include in event message	Use this parameter if you selected **Yes** in *Include deregistered phone details in event message*. Specify the maximum number of detail rows to include in an event message. Each row contains details for one phone. Rows are sorted by most recently disconnected phone. Specify "0" to include all rows. The default is 20 rows.

PhoneDeregistrations

Use this Knowledge Script to monitor phone deregistrations for Communication Manager and to maintain deregistration history in the Avaya CM supplemental database. This script raises an event if the number or percentage of deregistered phones exceeds the threshold you set. You can group events by building, floor, or type of phone.

NOTE: If you bypass SNMP to discover a call manager, this script is not available.

Prerequisites

- Run SetupSupplementalDB to create the Avaya CM supplemental database.
- Run RetrieveConfigData to populate the Avaya CM supplemental database.

Resource Object

AvayaCM Active SPE object

Default Schedule

By default, this script runs every 10 minutes.

Setting Parameter Values

Set the following parameters as needed:

Parameter	How to Set It
General Settings	
Job Failure Notification	
Event severity when job fails	Set the event severity level, from 1 to 40, to indicate the importance of the failure of the PhoneDeregistrations job. The default is 5.
Enable use of SNMP GETBulk operations?	By default, this parameter is enabled, allowing the PhoneDeregistrations job to use `GETNext` and `GETBulk` SNMP requests to access Communication Manager MIBs. Disable this parameter to allow the script to use only `GETNext` requests. Not all MIB tables are extensive enough to need a `GETBulk` request. A `GETBulk` request is faster, but more CPU-intensive than `GETNext`.
Number of rows to request for each GETBulk operation	Specify the number of rows from the MIB table to return in a `GETBulk` request. The default is 10 rows. The number of rows determines how quickly MIB data is returned. If CPU usage is too high, you can reduce the number of rows per `GETBulk` or disable the *Enable use of SNMP GETBulk requests?* parameter.
Interval to pause between GETBulk operations	Specify the number of milliseconds to wait between GETBulk requests. The default is 100 milliseconds. The length of delay can help with managing CPU usage and speed of SNMP requests. For example, a one-row `GETBulk` with a 100-millisecond delay between requests executes more slowly and uses less CPU than a `GETNext` request.
Event Notification	

Parameter	How to Set It
Raise event if deregistered phones in group exceed threshold?	Select **Yes** to raise an event if the number or percentage of deregistered phones in a group exceeds the threshold you set. The default is Yes. Use *Type of threshold* to create a “number” or “percentage” event. Use *Select event grouping* to select how to group the deregistered phones.
Select event grouping	Select whether to group deregistered phones by **Cluster**, **Building**, **Floor,** or **Type** of phone. AppManager raises an event when the number or percentage of deregistered phones in *each* group exceeds the threshold you set. For example, suppose the following were true: ◆ You set *Maximum number of deregistered phones in the group* to “5.” ◆ You set *Select event grouping* to “Building,” and you have three buildings. If AppManager detects six deregistered phones in the first building, two in the second, and seven in the third, it will raise two events: one for the six deregistered phones in the first building and another for the seven deregistered phones in the third building. Because you set the threshold to “5,” no event is raised for the deregistered phones in the second building. The default is Cluster.
Type of threshold	Select whether you want to raise events based on the **Number** or **Percent** of deregistered phones. The default is Number.
Threshold - Maximum number of deregistered phones	Use this parameter if you selected **Number** in *Type of threshold*. Specify the maximum number of phones that can be deregistered before an event is raised. The default is 0 phones.
Threshold - Maximum percent of deregistered phones	Use this parameter if you selected **Percent** in *Type of threshold*. Specify the maximum percentage of phones that can be deregistered before an event is raised. The default is 0%.
Event severity when deregistered phones exceed threshold	Set the event severity, from 1 to 40, to indicate the importance of an event in which the number or percentage of deregistered phones in a group exceeds the threshold you set. The default is 15.
Include deregistered phone details in event message?	Select **Yes** to include details of the deregistered phones in the event message. Phone details can include station extension, station name, building, floor, station type, room, cable, jack, port, and deregistration time. The default is Yes.
Maximum number of detail rows to include in event message	Use this parameter if you selected **Yes** in *Include deregistered phone details in event message*. Specify the maximum number of detail rows to include in an event message. Each row contains details for one phone. Rows are sorted by most recently deregistered phone. Specify “0” to include all rows. The default is 20 rows.

PhoneInventory

Use this Knowledge Script to create an inventory of the phones configured in a Communication Manager cluster. You choose both the search criteria for the inventory and the location of the output folder for the results file containing the inventory list. Unless you specify a UNC path,

*servername**sharename**directoryname**filename*, the results file is written to the computer on which the NetIQ AppManager agent is running. If you specify a UNC path, ensure the `NetIQmc` service is running as an account that has proper permissions on the UNC path.

NOTE: If you bypass SNMP to discover a call manager, this script is not available.

Resource Object

AvayaCM Active SPE object

Default Schedule

By default, this script runs once.

Setting Parameter Values

Set the following parameters as needed:

Parameter	How to Set It
General Settings	
Job Failure Notification	
Event severity when job fails	Set the event severity level, from 1 to 40, to indicate the importance of the failure of the PhoneInventory job. The default is 5.
Enable use of SNMP GETBulk operations?	By default, this parameter is enabled, allowing the PhoneInventory job to use `GETNext` and `GETBulk` SNMP requests to access Communication Manager MIBs. Disable this parameter to allow the script to use only `GETNext` requests. Not all MIB tables are extensive enough to need a `GETBulk` request. A `GETBulk` request is faster, but more CPU-intensive than `GETNext`.
Number of rows to request for each GETBulk operation	Specify the number of rows from the MIB table to return in a `GETBulk` request. The default is 10 rows. The number of rows determines how quickly MIB data is returned. If CPU usage is too high, you can reduce the number of rows per `GETBulk` or disable the *Enable use of SNMP GETBulk requests?* parameter.
Interval to pause between GETBulk operations	Specify the number of milliseconds to wait between GETBulk requests. The default is 100 milliseconds. The length of delay can help with managing CPU usage and speed of SNMP requests. For example, a one row `GETBulk` with a 100-millisecond delay between requests executes more slowly and uses less CPU than a `GETNext` request.

Parameter	How to Set It
Raise event if phone inventory succeeds?	Select **Yes** to raise an event when a phone inventory file is successfully generated. The default is Yes.
Event severity when phone inventory succeeds	Set the event severity level, from 1 to 40, to indicate the importance of an event in which an inventory file is successfully generated. The default is 25.
Raise event if no records found?	Select **Yes** to raise an event when the PhoneInventory job finds no phones based on the criteria you selected. The default is Yes.
Event severity when no records found	Set the event severity level, from 1 to 40, to indicate the importance of an event in which the PhoneInventory job found no phones based on the criteria you selected. The default is 25.
Search Options	
Select by	Select the filter by which you want to create the list of phones. ◆ Name ◆ Extension (the default) ◆ Building ◆ Floor ◆ Type
Selection criteria	Type the selection criteria for the phones to be listed. You can specify the actual item or you can specify a pattern by using the * wildcard. For example, to monitor all the phones in the ADM building, enter `ADM*`. You can enter multiple items by separating each item with a comma. For example: `ADM0009A*,ADM0009B*` The items you enter must be of the same type as the *Select by* parameter. So if *Select by* is Name, then the items you enter must be device names or patterns. If *Select by* is Extension, then the items you enter must be phone extension numbers. **NOTE:** Only the following characters are acceptable in this parameter: ◆ Number ◆ Uppercase and lowercase letters ◆ Periods ◆ Commas ◆ Asterisks (*) ◆ Underscores ◆ Spaces

Parameter	How to Set It
List only phones with status of	To further filter the list of phones, select a status. Only phones with this status that also match the criteria you specified in *Selection criteria* and *Select by* will be included in the inventory list. Select from the following status types: ◆ Any ◆ Registered ◆ Registered Not Connected ◆ Unregistered
Result File Options	
Full path to output folder for result file	Type the full path or a UNC path to a location on the agent computer in which to save the inventory `.csv` file. The default path is `c:\Program Files\NetIQ\Temp\NetIQ_Debug\PhoneInventory.csv`
Order by	Select **Name** to display the contents of the results file in order by phone name. Select **Extension** to display the contents of the results file in order by phone extension number. The default is Extension.

PhoneQuality

Use this Knowledge Script to collect real-time voice-quality statistics for active calls on Avaya IP phones. This script raises one event per call if statistics exceed or fall below the thresholds you set. In addition, this script generates data streams for the following statistics:

- Maximum interval MOS
- Maximum interval R-Value
  MOS and R-Value are computed only for calls that use one of the following codecs: G.711u, G.711a, or G.729.
- Maximum interval jitter
- Maximum interval latency
- Maximum interval packet loss

NOTE: You can trigger NetIQ Vivinet Diagnostics to diagnose the problem when voice quality thresholds are exceeded. For more information, see "Triggering Call and Phone Quality Diagnoses" on page 66.

When you start the PhoneQuality Knowledge Script job, the managed object begins collecting voice quality statistics. The managed object stops collecting statistics approximately one minute after the PhoneQuality job stops. If you attempt to delete a phone on which collection is still occurring, the following event is raised:

```
The phone(s) could not be removed because one or more phones are currently being
monitored by the PhoneQuality Knowledge Script. You must stop the PhoneQuality
job(s) before removing the phones.
```

NOTE: Unlike other proxy-based AppManager modules, AppManager for Avaya Communication Manager supports only one AppManager repository (`QDB`) per proxy agent computer. This limitation ensures the accuracy of monitoring phones with the PhoneQuality script. The list of phones for monitoring with this script does not differentiate between repositories; if multiple repositories were allowed, you could very well monitor the wrong set of phones for a given repository.

Prerequisite

Run AddPhone to add phones to be monitored. The PhoneQuality script is not available until after you run the AddPhone script.

Resource Object

AvayaCM Station object

NOTE: Do not monitor more than 100 active phone (station) objects in one cluster or across multiple clusters.

Default Schedule

By default, this script runs on an asynchronous schedule.

Setting Parameter Values

Set the following parameters as needed:

Description	How To Set It
General Settings	
Job Failure Notification	
Event severity when job fails	Set the event severity level, from 1 to 40, to indicate the importance of the failure of the PhoneQuality job. The default is 5.
Monitor Settings	
Data collection interval for voice quality metrics	Specify the interval at which data points are generated for voice quality metrics. The default is 30 seconds. The minimum is 15 seconds. **NOTE:** Communication Manager sends RTCP packets to the proxy computer in almost real-time. If your data collection interval is the default, 30 seconds, and you are monitoring 100 phones and collecting data for only one metric, such as MOS, AppManager will generate about three data points per second (100 phones / 30 seconds). With all five metrics enabled, AppManager will generate about 16 data points per second. If you change to a more frequent data collection interval, for example every 15 seconds, AppManager will generate about 32 data points per second.

Description	How To Set It
Additional fixed delay for MOS/R-Value calculation	Enter a length of delay (in milliseconds) that you want to add to a call. This delay is in addition to the three other types of delay associated with calculating MOS and R-Value: ◆ Network delay in one direction. ◆ Packetization delay. This value is fixed, based on the type of codec being used. ◆ Jitter buffer delay. This value is fixed, based on the type and size of the jitter buffer being used. The default is 0 milliseconds.
Monitor Interval MOS	
Event Notification	
Raise event if interval MOS falls below threshold?	Select **Yes** to raise an event if the value of interval MOS falls below the threshold that you set. The default is Yes. Interval MOS is the MOS value measured *during* the data collection interval you set. It is not the MOS value at the instance of data collection.
Threshold - Minimum interval MOS	Specify the lowest interval MOS value that can be calculated before an event is raised. The default is 3.60.
Event severity when interval MOS falls below threshold	Set the event severity level, from 1 to 40, to indicate the importance of an event in which the interval MOS value falls below the threshold you set. The default is 5.
Data Collection	
Collect data for interval MOS?	Select **Yes** to collect interval MOS data for charts and graphs. When enabled, data collection returns the value of interval MOS measured during the data collection interval. The default is Yes.
Monitor Interval R-Value	
Event Notification	
Raise event if interval R-Value falls below threshold?	Select **Yes** to raise an event if interval R-Value falls below the threshold that you set. The default is Yes. Interval R-Value is the R-Value measured *during* the data collection interval you set. It is not the R-Value at the instance of data collection.
Threshold - Minimum interval R-value	Enter the lowest interval R-Value that can be calculated before an event is raised. The default is 70.
Event severity when interval R-value falls below threshold	Set the event severity level, from 1 to 40, to indicate the importance of an event in which interval R-Value falls below the threshold you set. The default is 5.
Data Collection	
Collect data for interval R-Value?	Select **Yes** to collect interval R-Value data for charts and graphs. When enabled, data collection returns the value of interval R-Value measured during the data collection interval. The default is unselected.
Monitor Interval Jitter	
Event Notification	

Description	How To Set It
Raise event if interval jitter exceeds threshold?	Select **Yes** to raise an event if the amount of interval jitter exceeds the threshold that you set. The default is Yes. Interval jitter is the jitter value measured *during* the data collection interval you set. It is not the jitter value at the instance of data collection.
Threshold - Maximum interval jitter	Specify the highest amount of interval jitter that can be achieved before an event is raised. The default is 60 milliseconds.
Event severity when interval jitter exceeds threshold	Set the event severity level, from 1 to 40, to indicate the importance of an event in which the amount of interval jitter exceeds the threshold you set. The default is 15.
Data Collection	
Collect data for interval jitter?	Select **Yes** to collect interval jitter data for charts and graphs. When enabled, data collection returns the value of interval jitter measured during the data collection interval. The default is unselected.
Monitor Interval Latency	
Event Notification	
Raise event if interval latency exceeds threshold?	Select **Yes** to raise an event if the amount of interval latency exceeds the threshold that you set. The default is Yes. Interval latency is the latency value measured *during* the data collection interval you set. It is not the latency value at the instance of data collection.
Threshold - Maximum interval latency	Specify the highest amount of interval latency that can be achieved before an event is raised. The default is 400 milliseconds.
Event severity when interval latency exceeds threshold	Set the event severity level, from 1 to 40, to indicate the importance of an event in which the amount of interval latency exceeds the threshold you set. The default is 15.
Data Collection	
Collect data for interval latency?	Select **Yes** to collect interval latency data for charts and graphs. When enabled, data collection returns the value of interval latency measured during the data collection interval. The default is unselected.
Monitor Interval Packet Loss	
Event Notification	
Raise event if interval packet loss exceeds threshold?	Select **Yes** to raise an event if the percentage of interval packet loss exceeds the threshold that you set. The default is Yes. Interval packet loss is the percentage of packet loss measured *during* the data collection interval you set. It is not the packet loss value at the instance of data collection.
Threshold - Maximum interval packet loss	Specify the highest percentage of interval packet loss that can occur before an event is raised. The default is 1.0%.
Event severity when interval packet loss exceeds threshold	Set the event severity level, from 1 to 40, to indicate the importance of an event in which the percentage of interval packet loss exceeds the threshold you set. The default is 15.
Data Collection	

Description	How To Set It
Collect data for interval packet loss?	Select **Yes** to collect interval packet loss data for charts and graphs. When enabled, data collection returns the percentage of interval packet loss measured during the data collection interval. The default is unselected.

RegisteredResources

Use this Knowledge Script to monitor changes in the number of resources registered to a Communication Manager server. Resources include IP stations, remote office stations, attendant consoles, and H.248 media gateways. This script raises an event when a threshold is exceeded. In addition, this script generates data streams for the following statistics:

- Number of registered IP stations
- Percentage of increase and decrease in number of registered IP stations
- Number of registered IP attendant consoles
- Percentage of increase and decrease in number of registered IP attendant consoles
- Number of registered remote office stations
- Percentage of increase and decrease in number of registered remote office stations
- Number of registered H.248 media gateways
- Percentage of increase and decrease in number of registered H.248 media gateways

NOTE: If you bypass SNMP to discover a call manager, this script is not available.

Resource Object

AvayaCM Active SPE object

Default Schedule

By default, this script runs every 5 minutes.

Setting Parameter Values

Set the following parameters as needed:

Parameter	How to Set It
General Settings	
Job Failure Notification	
Event severity when job fails	Set the event severity level, from 1 to 40, to indicate the importance of the failure of the RegisteredResources job. The default is 5.

Parameter	How to Set It
Enable use of SNMP GETBulk operations?	Select **Yes** to allow this Knowledge Script to access Communication Manager MIBs using `GETNext` and `GETBulk` SNMP requests, as appropriate. The default is Yes. Deselect the check box to allow the Knowledge Script to use only `GETNext` requests. Not all MIB tables are extensive enough to need a `GETBulk` request. A `GETBulk` request is faster, but more CPU-intensive than a `GETNext` request.
Number of rows to request for each GETBulk operation	Specify the number of rows from the MIB table to return in a `GETBulk` request. The default is 10 rows. The number of rows determines how fast the MIB data is returned. If the CPU usage is too high, you can reduce the number of rows per `GETBulk` request or deselect the *Enable use of SNMP GETBulk requests?* parameter.
Interval to pause between GETBulk operations	Specify the number of milliseconds to wait between GETBulk requests. The default is 100 milliseconds. The length of the delay can help manage CPU usage and the speed of SNMP requests. For example, a one-row `GETBUlk` with a 100-millisecond delay between requests executes more slowly and uses less CPU than a `GETNext` request.
Monitor Registered IP Stations	
Data Collection	
Collect data for registered IP stations?	Select **Yes** to collect data for charts and reports. If enabled, data collection returns the number of IP stations that were registered during the monitoring period. The default is Yes.
Monitor Increase in Registered IP Stations	
Event Notification	
Raise event if increase in registered IP stations exceeds threshold?	Select **Yes** to raise an event if the percentage of increase in registered IP stations exceeds the threshold you set. The default is Yes.
Threshold - Maximum increase in registered IP stations	Specify the highest percentage of increase in registered IP stations that can occur before an event is raised. The default is 10%.
Event severity when increase in registered IP stations exceeds threshold	Set the event severity level, from 1 to 40, to indicate the importance of an event in which the percentage of increase in registered IP stations exceeds the threshold. The default is 15.
Data Collection	
Collect data for increase in registered IP stations?	Select **Yes** to collect data for charts and reports. If enabled, data collection returns the percentage of increase in registered IP stations during the monitoring period. The default is unselected.
Monitor Decrease in Registered IP Stations	
Event Notification	

Parameter	How to Set It
Raise event if decrease in registered IP stations exceeds threshold?	Select **Yes** to raise an event if the percentage of decrease in registered IP stations exceeds the threshold you set. The default is Yes.
Threshold - Maximum decrease in registered IP stations	Specify the highest percentage of decrease in registered IP stations that can occur before an event is raised. The default is 10%.
Event severity when decrease in registered IP stations exceeds threshold	Set the event severity level, from 1 to 40, to indicate the importance of an event in which the percentage of decrease in registered IP stations exceeds the threshold. The default is 15.
Data Collection	
Collect data for decrease in registered IP stations?	Select **Yes** to collect data for charts and reports. If enabled, data collection returns the percentage of decrease in registered IP stations during the monitoring period. The default is unselected.
Monitor Registered IP Attendant Consoles	
Data Collection	
Collect data for registered IP attendant consoles?	Select **Yes** to collect data for charts and reports. If enabled, data collection returns the number of IP attendant consoles that were registered during the monitoring period. The default is Yes.
Monitor Increase in Registered IP Attendant Consoles	
Event Notification	
Raise event if increase in registered IP attendant consoles exceeds threshold?	Select **Yes** to raise an event if the percentage of increase in registered IP attendant consoles exceeds the threshold you set. The default is Yes.
Threshold - Maximum increase in registered IP attendant consoles	Specify the highest percentage of increase in registered IP attendant consoles that can occur before an event is raised. The default is 10%.
Event severity when increase in registered IP attendant consoles exceeds threshold	Set the event severity level, from 1 to 40, to indicate the importance of an event in which the percentage of increase in registered IP attendant consoles exceeds the threshold. The default is 15.
Data Collection	
Collect data for increase in registered IP attendant consoles?	Select **Yes** to collect data for charts and reports. If enabled, data collection returns the percentage of increase in registered IP attendant consoles during the monitoring period. The default is unselected.
Monitor Decrease in Registered IP Attendant Consoles	
Event Notification	
Raise event if decrease in registered IP attendant consoles exceeds threshold?	Select **Yes** to raise an event if the percentage of decrease in registered IP attendant consoles exceeds the threshold you set. The default is Yes.
Threshold - Maximum decrease in registered IP attendant consoles	Specify the highest percentage of decrease in registered IP attendant consoles that can occur before an event is raised. The default is 10%.

Parameter	How to Set It
Event severity when decrease in registered IP attendant consoles exceeds threshold	Set the event severity level, from 1 to 40, to indicate the importance of an event in which the percentage of decrease in registered IP attendant consoles exceeds the threshold. The default is 15.
Data Collection	
Collect data for decrease in registered IP attendant consoles?	Select **Yes** to collect data for charts and reports. If enabled, data collection returns the percentage of decrease in registered IP attendant consoles during the monitoring period. The default is unselected.
Monitor Registered Remote Office Stations	
Data Collection	
Collect data for registered remote office stations?	Select **Yes** to collect data for charts and reports. If enabled, data collection returns the number of remote office stations that were registered during the monitoring period. The default is Yes.
Monitor Increase in Registered Remote Office Stations	
Event Notification	
Raise event if increase in registered remote office stations exceeds threshold?	Select **Yes** to raise an event if the percentage of increase in registered remote office stations exceeds the threshold you set. The default is Yes.
Threshold - Maximum increase in registered remote office stations	Specify the highest percentage of increase in registered remote office stations that can occur before an event is raised. The default is 10%.
Event severity when increase in registered remote office stations exceeds threshold	Set the event severity level, from 1 to 40, to indicate the importance of an event in which the percentage of increase in registered remote office stations exceeds the threshold. The default is 15.
Data Collection	
Collect data for increase in registered remote office stations?	Select **Yes** to collect data for charts and reports. If enabled, data collection returns the percentage of increase in registered remote office stations during the monitoring period. The default is unselected.
Monitor Decrease in Registered Remote Office Stations	
Event Notification	
Raise event if decrease in registered remote office stations exceeds threshold?	Select **Yes** to raise an event if the percentage of decrease in registered remote office stations exceeds the threshold you set. The default is Yes.
Threshold - Maximum decrease in registered remote office stations	Specify the highest percentage of decrease in registered remote office stations that can occur before an event is raised. The default is 10%.
Event severity when decrease in registered remote office stations exceeds threshold	Set the event severity level, from 1 to 40, to indicate the importance of an event in which the percentage of decrease in registered remote office stations exceeds the threshold. The default is 15.
Data Collection	

Parameter	How to Set It
Collect data for decrease in registered remote office stations?	Select **Yes** to collect data for charts and reports. If enabled, data collection returns the percentage of decrease in registered remote office stations during the monitoring period. The default is unselected.
Monitor Registered H.248 Media Gateways	
Data Collection	
Collect data for registered H.248 media gateways?	Select **Yes** to collect data for charts and reports. If enabled, data collection returns the number of H.248 media gateways that were registered during the monitoring period. The default is Yes.
Monitor Increase in Registered H.248 Media Gateways	
Event Notification	
Raise event if increase in registered H.248 media gateways exceeds threshold?	Select **Yes** to raise an event if the percentage of increase in registered H.248 media gateways exceeds the threshold you set. The default is Yes.
Threshold - Maximum increase in registered H.248 media gateways	Specify the highest percentage of increase in registered H.248 media gateways that can occur before an event is raised. The default is 10%.
Event severity when increase in registered H.248 media gateways exceeds threshold	Set the event severity level, from 1 to 40, to indicate the importance of an event in which the percentage of increase in registered H.248 media gateways exceeds the threshold. The default is 15.
Data Collection	
Collect data for increase in registered H.248 media gateways?	Select **Yes** to collect data for charts and reports. If enabled, data collection returns the percentage of increase in registered H.248 media gateways during the monitoring period. The default is unselected.
Monitor Decrease in Registered H.248 Media Gateways	
Event Notification	
Raise event if decrease in registered H.248 media gateways exceeds threshold?	Select **Yes** to raise an event if the percentage of decrease in registered H.248 media gateways exceeds the threshold you set. The default is Yes.
Threshold - Maximum decrease in registered H.248 media gateways	Specify the highest percentage of decrease in registered H.248 media gateways that can occur before an event is raised. The default is 10%.
Event severity when decrease in registered H.248 media gateways exceeds threshold	Set the event severity level, from 1 to 40, to indicate the importance of an event in which the percentage of decrease in registered H.248 media gateways exceeds the threshold. The default is 15.
Data Collection	
Collect data for decrease in registered H.248 media gateways?	Select **Yes** to collect data for charts and reports. If enabled, data collection returns the percentage of decrease in registered H.248 media gateways during the monitoring period. The default is unselected.

RemovePhone

Use this Knowledge Script to remove Avaya IP phone objects from the TreeView in the AppManager console. This script raises an event when phones are removed successfully.

TIP

- After running this script on the object you want to remove, double-check your selection in the Objects tab. By specifically selecting a phone object from the Objects tab, you will not accidentally remove a phone that you want to keep.
- Before removing a phone, stop any monitoring jobs that are running on the phone.

Resource Object

AvayaCM Station object

Default Schedule

By default, this script runs once.

Setting Parameter Values

Set the following parameters as needed:

Parameter	How To Set It
General Settings	
Job Failure Notification	
Event severity when job fails	Set the event severity level, from 1 to 40, to indicate the importance of the failure of the RemovePhone job. The default is 5.
Event Notification	
Raise event if phones are removed successfully?	Select **Yes** to raise an event if the selected phone objects are successfully removed from the TreeView. The default is Yes.
Event severity when phones are removed successfully	Set the event severity level, from 1 to 40, to indicate the importance of an event in which the selected phone objects are successfully removed from the TreeView. The default is 25.

RetrieveConfigData

Use this Knowledge Script to retrieve configuration data about stations and gateways from the Communication Manager server and store it in the Avaya CM supplemental database.

NOTE: If you bypass SNMP to discover a call manager, this script is not available.

Prerequisite

Create the supplemental database using the SetupSupplementalDB script or the *Set up supplemental database?* parameters in the Discovery_AvayaCM Knowledge Script.

Resource Object

AvayaCM Active SPE object

Default Schedule

By default, this script runs once a day, at 3 AM, in order to perform CPU-intensive functions when Communication Manager is less busy with call activity or maintenance tasks.

However, because the PhoneDeregistrations script uses the configuration data this script retrieves, you might want to set this script to “Run Once” so the configuration data is retrieved immediately. When the “Run Once” job is complete, you can then run this script using the default schedule of once daily.

Setting Parameter Values

Set the following parameters as needed:

Parameter	How to Set It
General Settings	
Job Failure Notification	
Event severity when job fails	Set the event severity level, from 1 to 40, to indicate the importance of the failure of the RetrieveConfigData job. The default is 5.
Enable use of SNMP GETBulk operations?	By default, this parameter is enabled, allowing the RetrieveConfigData Knowledge Script job to use `GETNext` and `GETBulk` SNMP requests to access Communication Manager MIBs. Disable this parameter to allow the script to use only `GETNext` requests. Not all MIB tables are extensive enough to need a `GETBulk` request. A `GETBulk` request is faster, but more CPU-intensive than `GETNext`.
Number of rows to request for each GETBulk operation	Specify the number of rows from the MIB table to return in a `GETBulk` request. The default is 10 rows. The number of rows determines how quickly MIB data is returned. If CPU usage is too high, you can reduce the number of rows per `GETBulk` or disable the *Enable use of SNMP GETBulk requests?* parameter.
Interval to pause between GETBulk operations	Specify the number of milliseconds to wait between GETBulk requests. The default is 100 milliseconds. The length of delay can help with managing CPU usage and speed of SNMP requests. For example, a one-row `GETBulk` with a 100-millisecond delay between requests executes more slowly and uses less CPU than a `GETNext` request.

Parameter	How to Set It
Raise event if configuration retrieval succeeds?	Select **Yes** to raise an event if the retrieval process succeeds. The default is unselected.
Event severity when configuration retrieval succeeds	Set the event severity level, from 1 to 40, to indicate the importance of an event in which retrieval succeeds. The default is 25.

SecurityViolations

Use this Knowledge Script to monitor the following security violations:

- Barrier code violations
- Calls that generated authorization code violations
- Calls that generated station security code violations

NOTE: If you bypass SNMP to discover a call manager, this script is not available.

Barrier codes and authorization codes provide a level of security for remote call access to such telephony components as PBXs, switch features, and trunks. Station security codes enable the Personal Station Access feature and the Extended User Administration of Redirected Calls feature.

This script raises an event if a threshold is exceeded. In addition, this script generates data streams for monitored violations.

Resource Object

AvayaCM Active SPE object

Default Schedule

By default, this script runs every 5 minutes.

Setting Parameter Values

Set the following parameters as needed:

Parameter	How to Set It
General Settings	
Job Failure Notification	
Event severity when job fails	Set the event severity level, from 1 to 40, to indicate the importance of the failure of the SecurityViolations job. The default is 5.

Parameter	How to Set It
Select port type to monitor	Select the type of login port you want to monitor for security violations. Choose from the following: ◆ All (default) ◆ SYSAM-LCL (local port) ◆ SYSAM-RMT (remote port) ◆ MAINT (maintenance port) ◆ SYS-Port (system port) ◆ MGR1 (management terminal connection port) ◆ NET (network controller port) ◆ EPN (EPN maintenance EIA port) ◆ INADS (initialization administration system port)
Enable use of SNMP GETBulk operations?	By default, this parameter is enabled, allowing the SecurityViolations Knowledge Script job to use `GETNext` and `GETBulk` SNMP requests to access Communication Manager MIBs. Disable this parameter to allow the script to use only `GETNext` requests. Not all MIB tables are extensive enough to need a `GETBulk` request. A `GETBulk` request is faster, but more CPU-intensive than `GETNext`.
Number of rows to request for each GETBulk operation	Specify the number of rows from the MIB table to return in a `GETBulk` request. The default is 10 rows. The number of rows determines how quickly MIB data is returned. If CPU usage is too high, you can reduce the number of rows per `GETBulk` or disable the *Enable use of SNMP GETBulk requests?* parameter.
Interval to pause between GETBulk operations	Specify the number of milliseconds to wait between GETBulk requests. The default is 100 milliseconds. The length of delay can help with managing CPU usage and speed of SNMP requests. For example, a one-row `GETBulk` with a 100-millisecond delay between requests executes more slowly and uses less CPU than a `GETNext` request.
Monitor Violations for Barrier Codes	
Event Notification	
Raise event if number of violations for barrier codes exceeds threshold?	Select **Yes** to raise an event if the number of security violations for barrier codes exceeds the threshold you set. The default is Yes.
Threshold - Maximum violations for barrier codes	Specify the highest number of security violations for barrier codes that can occur before an event is raised. The default is 0 violations.
Event severity when number of violations for barrier codes exceeds threshold	Set the event severity level, from 1 to 40, to indicate the importance of an event in which the number of security violations for barrier codes exceeds the threshold. The default is 15.
Data Collection	

Parameter	How to Set It
Collect data for number of violations for barrier codes?	Select **Yes** to collect data for charts and reports. If enabled, data collection returns the number of security violations for barrier codes that occurred during the monitoring period. The default is unselected.
Monitor Violations for Authorization Codes	
Event Notification	
Raise event if number of calls that generated authorization code violations exceeds threshold?	Select **Yes** to raise an event if the number of calls that generated authorization code violations exceeds the threshold you set. The default is Yes.
Threshold - Maximum number of calls that generated authorization code violations	Specify the highest number of calls that can generate authorization code violations before an event is raised. The default is 0 calls.
Event severity when the number of calls that generated authorization code violations exceeds threshold	Set the event severity level, from 1 to 40, to indicate the importance of an event in which the number of calls that generated authorization code violations exceeds the threshold. The default is 15.
Data Collection	
Collect data for the number of calls that generated authorization code violations?	Select **Yes** to collect data for charts and reports. If enabled, data collection returns the number of that generated authorization code violations during the monitoring period. The default is unselected.
Monitor Station Violations	
Event Notification	
Raise event if the number of calls that generated station violations exceeds threshold?	Select **Yes** to raise an event if the number of calls that generated station violations exceeds the threshold you set. The default is Yes.
Threshold - Maximum number of calls that generated station violations	Specify the highest number of calls that can generate station violations before an event is raised. The default is 0 calls.
Event severity when the number of calls that generated station violations exceeds threshold	Set the event severity level, from 1 to 40, to indicate the importance of an event in which the number of calls that generated station violations exceeds the threshold. The default is 15.
Data Collection	
Collect data for the number of calls that generated station violations?	Select **Yes** to collect data for charts and reports. If enabled, data collection returns the number of calls that generated station violations during the monitoring period. The default is unselected.

SetupSupplementalDB

Use this Knowledge Script to create an Avaya CM supplemental database, including the tables and stored procedures needed to store call detail records (CDRs), disconnected phone information, and deregistered phone information.

You can also create the Avaya CM supplemental database using the *Set up supplemental database?* parameters in the Discovery_AvayaCM Knowledge Script.

For more information, see "Understanding the Avaya CM Supplemental Database" on page 102.

Resource Object

AvayaCM Active SPE object

Default Schedule

By default, this script runs once.

Setting Parameter Values

Set the following parameters as needed:

Parameter	How to Set It
General Settings	
Job Failure Notification	
Event severity when job fails	Set the event severity level, from 1 to 40, to indicate the importance of the failure of the SetupSupplementalDB job. The default is 5.
Raise event if database setup succeeds?	Select **Yes** to raise an event if creation of the Avaya CM supplemental database is successful. The default is unselected.
Event severity when database setup succeeds	Set the event severity level, from 1 to 40, to indicate the importance of the success of the creation of the Avaya CM supplemental database. The default is 25.
CDR Parameters	
Number of days to keep call detail records	Specify the number of days' worth of CDRs you want to keep in the Avaya CM supplemental database. Data older than what you specify is discarded. The default is 7 days. **NOTE:** Run DeleteCDRRecords Knowledge Script to remove old records from the supplemental database.
SQL Server Information	
Local SQL Server instance name	Specify the name of the local SQL Server instance (on the proxy agent computer) in which you want to create the new Avaya CM supplemental database. Leave this parameter blank to accept the default name.

Understanding the Avaya CM Supplemental Database

The Avaya CM supplemental database is a Microsoft SQL Server database you create on the proxy agent computer. The supplemental database fulfills several functions.

Storage for CDRs and RTCP packets

The managed object on the proxy agent computer receives call detail records (CDRs) from Communication Manager servers and RTCP packets from phones registered to Communication Manager servers. The proxy agent computer saves the CDR and RTCP packet data to tables in the Avaya CM supplemental database. The CallActivity, CallFailures, CallQuality, and CallQuery Knowledge Scripts query the supplemental database for the data they need.

When you start the CallActivity, CallFailures, CallQuality, or CallQuery Knowledge Script job, the managed object begins collecting CDR and RTCP data to store in the Avaya CM supplemental database. After the job stops, the managed object continues to collect CDRs and packet data. Data collection stops within a time period equal to two intervals of the job, but never less than 4 minutes after the job stops.

When you create the supplemental database, you specify how long data is retained before being deleted and archived. AppManager automatically deletes CDRs older than the retention age you specify.

Storage for phone deregistration and disconnection data

The PhoneDeregistrations and PhoneConnectivity Knowledge Scripts use SNMP queries to create lists of phones that are unregistered or registered but disconnected. The scripts query the Avaya CM supplemental database, which is populated by the RetrieveConfigData script with configuration data retrieved from Communication Manager.

To create and use the supplemental database:

1. **Create the database**. Create one Avaya CM supplemental database per Communication Manager cluster you are monitoring. Use the Discovery_AvayaCM or SetupSupplementalDB Knowledge Script for this purpose.
2. **Populate the database**. Use RetrieveConfigData to retrieve configuration data from Communication Manager and save it in the Avaya CM supplemental database.
3. **Monitor the data in the database**. Use the following scripts to analyze the data in the database.
   - CallActivity monitors active and completed calls.
   - CallFailures monitors calls that ended with the condition codes you specify.
   - CallQuality monitors jitter, latency, lost data, and MOS.
   - CallQuery searches for data based on query filters you select.
   - PhoneConnectivity monitors disconnected registered phones and maintains a history of monitored phones in the supplemental database.
   - PhoneDeregistrations monitors phone deregistrations and maintains a history of phone deregistrations in the supplemental database.

SNMPTrap

Use this Knowledge Script to monitor SNMP traps forwarded from NetIQ SNMP Trap Receiver (Trap Receiver). This script raises an event when an SNMP trap is received and when Trap Receiver is unavailable or subsequently becomes available. In addition, this script generates data streams for Trap Receiver availability.

This script checks for SNMP traps in the MIB tree. You can add Management Information Bases (MIBs) to the MIB tree. For more information, see the AddMIB Knowledge Script.

Trap Receiver receives SNMP traps, filters them, and then forwards the traps to AppManager. For more information, see "Working with NetIQ SNMP Trap Receiver" on page 105.

Resource Object

AvayaCM Trap Receiver object

Default Schedule

By default, this script runs on an asynchronous schedule.

Setting Parameter Values

Set the following parameters as needed:

Parameter	How to Set It
Trap Filters	
List of trap OIDs	Use this parameter to provide a list of the OIDs (object identifiers) of the traps you want to monitor. Separate multiple OIDs with a comma. For example: `1.3.6.1.2.1.2.2.1.1.1,1.3.6.1.2.1.2.2.1.7.1`

Full path to file with list of trap OIDs	If you have many OIDs to monitor, use this parameter to identify the full path to a file that contains a list of the OIDs. Each OID in the file should be on a separate line. For example:
	`1.3.6.1.2.1.2.2.1.1.1`
	`1.3.6.1.2.1.2.2.1.7.1`
	Because the file must be accessible from the AppManager agent, the path must be a local directory on the agent computer or a UNC path. The `netiqmc` service must be running as a user that has access to the UNC path.

List of MIB subtrees	Use this parameter to monitor an OID *and* all of its subtrees. Provide a comma-separated list of the OIDs you want to monitor. For example: `1.3.6,1.3.7`

Parameter	How to Set It

Full path to file with list of MIB subtrees	If you have many subtrees to monitor, use this parameter to provide the full path to a file that contains a list of the OIDs. Each OID in the file should be on a separate line. For example:
	`1.3.6`
	`1.3.7`
	Because the file must be accessible from the AppManager agent, the path must be a local directory on the agent computer or a UNC path. The `netiqmc` service must be running as a user that has access to the UNC path.

Event Notification	
Format trap data according to SNMP version	Select the version of SNMP whose formatting should be used for trap event messages. The data provided by each format is the same; only the layout is different.
Include prefix information to format event messages for Netcool adapter?	Select **Yes** to format trap messages for use by IBM Tivoli Netcool. When this option is enabled, trap messages include tokens and separators, such as tildes (~), that Netcool recognizes.
Raise cleared/resolved alarm event?	Select **Yes** to raise an event when the SNMP trap message contains information about a cleared or resolved alarm. The default is Yes.
Event severity when cleared/ resolved alarm received	Set the severity level, from 1 to 40, to indicate the importance of an event in which the SNMP trap message contains information about a cleared or resolved alarm. The default is 25.
Raise critical alarm event?	Select **Yes** to raise an event when the SNMP trap message contains information about a critical alarm. The default is Yes.
Event severity when critical alarm received	Set the severity level, from 1 to 40, to indicate the importance of an event in which the SNMP trap message contains information about a critical alarm. The default is 5.
Raise major alarm event?	Select **Yes** to raise an event when the SNMP trap message contains information about a major alarm. The default is Yes.
Event severity when major alarm received	Set the severity level, from 1 to 40, to indicate the importance of an event in which the SNMP trap message contains information about a major alarm. The default is 10.
Raise minor alarm event?	Select **Yes** to raise an event when the SNMP trap message contains information about a minor alarm. The default is Yes.
Event severity when minor alarm received	Set the severity level, from 1 to 40, to indicate the importance of an event in which the SNMP trap message contains information about a minor alarm. The default is 15.
Raise warning alarm event?	Select **Yes** to raise an event when the SNMP trap message contains information about a warning alarm. The default is Yes.
Event severity when warning alarm received	Set the severity level, from 1 to 40, to indicate the importance of an event in which the SNMP trap message contains information about a warning alarm. The default is 15.
Raise unmapped alarm event?	Select **Yes** to raise an event when an SNMP trap is received but is not reflected in the `.CSV` mapping file. The default is Yes. Disable this parameter if you do not want to be informed about SNMP traps that are not mapped in the `.CSV` file.

Parameter	How to Set It
Event severity when unmapped alarm received	Set the severity level, from 1 to 40, to indicate the importance of an event in which an SNMP trap is not mapped in the `.CSV` file. The default is 15.
Raise Trap Receiver availability events?	Select **Yes** to raise an event when Trap Receiver becomes unavailable and when Trap Receiver becomes available once again. The default is Yes.
Event severity when Trap Receiver is unavailable	Set the severity level, from 1 to 40, to indicate the importance of an event in which Trap Receiver becomes unavailable. The default is 5.
Event severity when Trap Receiver becomes available	Set the severity level, from 1 to 40, to indicate the importance of an event in which Trap Receiver becomes available after being unavailable. The default is 25.
Data Collection	
Collect data for Trap Receiver availability?	Select **Yes** to collect data for charts and reports. If enabled, data collection returns “1” if Trap Receiver is available and “0” if Trap Receiver is unavailable. The default is unselected.
Interval for collecting Trap Receiver availability data	Specify the frequency with which the script collects Trap Receiver availability data. The default is every 5 minutes.

Working with NetIQ SNMP Trap Receiver

Installation of AppManager for Avaya Communication Manager automatically installs NetIQ SNMP Trap Receiver (Trap Receiver), which runs as a service: `NetIQTrapReceiver.exe`. Trap Receiver may compete for port usage with any other trap receiver installed on the same computer.

What is NetIQ SNMP Trap Receiver?

In general, a trap receiver is an application that receives traps from SNMP agents. Trap Receiver receives, filters, and forwards SNMP traps to AppManager. When you use Trap Receiver with AppManager for Avaya Communication Manager, the SNMPTrap Knowledge Script raises events when SNMP traps are received.

What is an SNMP Trap?

Simple Network Management Protocol (SNMP) is a protocol-based system used to manage devices on TCP/IP-based networks. From devices on which an SNMP agent resides, such as routers and switches, SNMP sends unsolicited notifications, called traps, to network administrators when thresholds for certain conditions are exceeded. These conditions are defined by the vendor in a device’s Management Information Base (MIB); the network administrator sets the thresholds.

Traps are composed of Protocol Data Units (PDUs). Each PDU contains the following information, organized in various ways depending on the version of SNMP in use:

- SNMP version number
- Community name of the SNMP agent
- PDU type
- Enterprise OID (object identifier), a unique number that identifies an enterprise and its system objects in the MIB
- IP address of the SNMP agent

- Generic trap type: Cold start, Warm start, Link down, Link up, Authentication failure, and Enterprise
- Specific trap type. When the Generic trap type is set to “Enterprise,” a specific trap type is included in the PDU. A specific trap is one that is unique or specific to an enterprise.
- Time the event occurred
- Varbind (variable binding), a sequence of two fields that contain the OID and a value

Understanding Trap Receiver Architecture

Trap Receiver operates on a Client-Server architecture: the *Server*—the stand-alone Trap Receiver application—receives, filters, and forwards SNMP traps to the *Client*—an application that receives traps, such as AppManager. The Server may receive traps from standard UDP port 162 or from any other configured port. The Client and the Server can reside on the same computer or on separate (proxy) computers.

Communication between Client and Server is implemented as XML messages over a TCP connection. Only one Server is allowed per computer; however, several Clients are allowed per computer. Clients that are registered to the same Server share the same TCP connection. The Server TCP port should be known to all potential Clients.

Understanding the Trap Receiver Configuration File

The configuration file for Trap Receiver, `NetIQTrapReceiver.conf`, identifies the UDP and TCP ports used by Trap Receiver: the UDP port is used for receiving traps; the TCP port is used for communicating with the Client, such as AppManager or another supported NetIQ application. The configuration file also identifies the level of logging you want to use and whether port forwarding is enabled.

By default, the configuration file is installed in [*installation directory*]`\config`, and has the following format:

```
##############################################################
## NetIQTrapReceiver.conf
# A configuration file for NetIQ SNMP Trap Receiver
#################################################################
#########################
# TCP port
# Syntax: tcp_port [port]
# E.g. : tcp_port 2735
#########################
tcp_port 2735
#########################
# UDP port
# Syntax: udp_port [port]
# E.g. : udp_port 162
#########################
udp_port 162
#########################
# Forwarding
# Syntax: forward [address]:[port] [v1]
# E.g. : forward 127.0.0.1:1000 v1
#########################
#########################
# Log level
# Syntax: log_level error|warning|info|debug|xml
# E.g. : log_level info
#########################
log_level debug
```

If the configuration file cannot be found, cannot be parsed, or does not contain one of the required values, Trap Receiver is initialized with the default configuration as shown above.

When changing values in the configuration file, take into account the following:

- If you change the TCP port number, stop all asynchronous Knowledge Script jobs associated with the modules that support Trap Receiver. Run the Discovery Knowledge Script on all monitored devices to enable the devices to recognize the new TCP port number.
- If you change the UDP port number, also change the UDP port number configured on the devices that send traps to Trap Receiver.
- If another service uses port 2735 or port 162, Trap Receiver *will not start*. The Trap Receiver log file will contain different levels of messages, based on the log_level you choose. Either change the port numbers in the configuration file, stop the service that is using the default Trap Receiver port numbers, or forward the traps coming in to UDP port 162.
- To forward incoming traps to another trap receiver, such as Microsoft SNMP Trap Service, set the Forwarding values as follows:
  `forward` [*IP address of other trap receiver*]`:`[*port number of other trap receiver*] [*SNMP*

version].
For example: `forward 10.40.40.25:167 v1`. By default, incoming traps are not forwarded. For more information, see "Coexisting with Microsoft SNMP Trap Service" on page 108.

- Restart Trap Receiver after any change to the configuration file. From Control Panel, double-click **Administrative Tools** and then double-click **Services**. Right-click **NetIQ Trap Receiver** and select **Restart**.

Coexisting with Microsoft SNMP Trap Service

Two trap receivers cannot be in use on the same computer while using the same standard UDP port (162). If NetIQ SNMP Trap Receiver and another trap receiver such as Microsoft SNMP Trap Service are installed on the same computer and both are receiving traps, then configure Trap Receiver to use the standard UDP port and to forward incoming traps (UDP forwarding) to the other trap receiver. For more information, see "Understanding the Trap Receiver Configuration File" on page 107.

Then, configure the other trap receiver to use a different, non-standard, UDP port that is not in use by another application. The following are instructions for configuring Microsoft SNMP Trap Service.

To configure Microsoft SNMP Trap Service to use another port:

1. Navigate to `c:\Windows\system32\drivers\etc`.
2. Open the **services** file.
3. In the row for `snmptrap`, change the value for **udp** from 162 to another port number that is not in use by any other application. Use the same port number you set as the forwarding port in the Trap Receiver configuration file.
4. Save and close the **services** file.
5. Restart Windows SNMP Trap Service. In Control Panel, double-click **Administrative Tools** and then double-click **Services**. Right-click **SNMP Trap Service** and select **Restart**.

TIP: To see which ports are in use, run `netstat.exe` from a command prompt. Then select an available port as the port for the other trap receiver service.

SystemUptime

Use this Knowledge Script to monitor the number of hours that an Avaya Communication Manager has been operational since its last reboot. This Knowledge Script creates an event if the server reboots. In addition, this script generates a data stream for server uptime. This Knowledge Script does not create an event when a monitored system fails; it only creates an event after the system reboots. Use the General_PingMachine Knowledge Script or the NetworkDevice_Device_Ping Knowledge Script, if you want to monitor when the system goes down.

NOTE: If you bypass SNMP to discover a call manager (by selecting the **Discover using manual configuration?** parameter), this script is not available.

Resource Object

AvayaCM Server object

Default Schedule

By default, this script runs every 5 minutes.

Setting Parameter Values

Set the following parameters as needed:

Parameter	How to Set It
General Settings	
Job Failure Notification	
Event severity when job fails	Set the event severity level, from 1 to 40, to indicate the importance of the failure of the SystemUptime job. The default is 5.
Monitor Reboot Events	
Event Notification	
Raise event if server reboots?	Select **Yes** to raise an event if the Avaya Communication Manager server reboots. The default is Yes.
Event severity when server reboots	Set the event severity level, from 1 to 40, to indicate the importance of an event in which the Avaya Communication Manager server reboots. The default is 5.
Monitor Uptime	
Data Collection	
Collect data for uptime?	Select **Yes** to collect data for charts and reports. If enabled, data collection returns the amount of time the Avaya Communication Manager server has been operational since its last reboot. The default is Yes.

TrunkGroupUsage

Use this Knowledge Script to monitor the status of a trunk group. This script raises an event if a monitored value exceeds the threshold you set. In addition, this script generates data streams for the following statistics:

- Total number of hours all trunks are busy with calls
- Percentage of time all trunks are simultaneously in use
- Calls in queue
- Calls not in queue
- Out-of-service trunks

NOTE: If you bypass SNMP to discover a call manager (by selecting the **Discover using manual configuration?** parameter), this script is not available.

Resource Object

AvayaCM Trunk Group object

Default Schedule

By default, this script runs every hour because the SNMP data it monitors is updated only once an hour. If you change the schedule to a shorter interval, you may receive SNMP request errors until the SNMP data is repopulated.

Setting Parameter Values

Set the following parameters as needed:

Parameter	How to Set It
General Settings	
Job Failure Notification	
Event severity when job fails	Set the event severity level, from 1 to 40, to indicate the importance of the failure of the TrunkGroupUsage job. The default is 5.
Enable use of SNMP GETBulk operations?	Select **Yes** to allow this Knowledge Script to access Communication Manager MIBs using `GETNext` and `GETBulk` SNMP requests, as appropriate. The default is Yes. Deselect the check box to allow the Knowledge Script to use only `GETNext` requests. Not all MIB tables are extensive enough to need a `GETBulk` request. A `GETBulk` request is faster, but more CPU-intensive than a `GETNext` request.
Number of rows to request for each GETBulk operation	Specify the number of rows from the MIB table to return in a `GETBulk` request. The default is 10 rows. The number of rows determines how fast the MIB data is returned. If the CPU usage is too high, you can reduce the number of rows per `GETBulk` request or deselect the *Enable use of SNMP GETBulk requests?* parameter.
Interval to pause between GETBulk operations	Specify the number of milliseconds to wait between GETBulk requests. The default is 100 milliseconds. The length of the delay can help manage CPU usage and the speed of SNMP requests. For example, a one-row `GETBUlk` with a 100-millisecond delay between requests executes more slowly and uses less CPU than a `GETNext` request.
Monitor Total Time Trunks Busy with Calls	
Event Notification	
Raise event if total hours trunks busy with calls exceeds threshold?	Select **Yes** to raise an event if the total number of hours that all trunks are busy with calls exceeds the threshold you set. The default is Yes.
Threshold - Maximum total hours trunks busy with calls	Specify the highest number of hours that all trunks can be busy with calls before an event is raised. The default is 1 hour.
Event severity when total hours trunks busy with calls exceeds threshold	Set the event severity level, from 1 to 40, to indicate the importance of an event in which the total number of hours that all trunks are busy with calls exceeds the threshold. The default is 15.
Data Collection	

Parameter	How to Set It
Collect data for total hours trunks busy with calls?	Select **Yes** to collect data for charts and reports. If enabled, data collection returns the total number of hours that all trunks were busy with calls during the monitoring period. The default is Yes.
Monitor Calls Queued	
Event Notification	
Raise event if number of calls queued exceeds threshold?	Select **Yes** to raise an event if the number of calls in queue exceeds the threshold you set. The default is Yes.
Threshold - Maximum calls queued	Specify the maximum number of calls that can be in queue before an event is raised. The default is 10 calls.
Event severity when number of calls queued exceeds threshold	Set the event severity level, from 1 to 40, to indicate the importance of an event in which the number of calls in queue exceeds the threshold. The default is 15.
Data Collection	
Collect data for calls queued?	Select **Yes** to collect data for charts and reports. If enabled, data collection returns the number of calls in queue during the monitoring period. The default is Yes.
Monitor Calls Not Queued	
Event Notification	
Raise event if number of calls not queued exceeds threshold?	Select **Yes** to raise an event if the number of calls not in queue exceeds the threshold you set. The default is Yes. Calls not in queue are calls that were offered to the trunk group when the queue was full.
Threshold - Maximum calls not queued	Specify the maximum number of calls that can be not queued before an event is raised. The default is 5 calls.
Event severity when number of calls not queued exceeds threshold	Set the event severity level, from 1 to 40, to indicate the importance of an event in which the number of calls not queued exceeds the threshold. The default is 15.
Data Collection	
Collect data for calls not queued?	Select **Yes** to collect data for charts and reports. If enabled, data collection returns the number of calls not queued during the monitoring period. The default is unselected.
Monitor Trunks Out of Service	
Event Notification	
Raise event if number of trunks out of service exceeds threshold?	Select **Yes** to raise an event if the number of out-of-service trunks exceeds the threshold you set. The default is Yes.
Threshold - Maximum trunks out of service	Specify the maximum number of trunks that can be out of service before an event is raised. The default is 1 trunk.
Event severity when number of trunks out of service exceeds threshold	Set the event severity level, from 1 to 40, to indicate the importance of an event in which the number of out-of-service trunks exceeds the threshold. The default is 15.
Data Collection	

Parameter	How to Set It
Collect data for trunks out of service?	Select **Yes** to collect data for charts and reports. If enabled, data collection returns the number of trunks that were out of service during the monitoring period. The default is Yes.
Monitor Percent Time Trunks in Use	
Event Notification	
Raise event if percent of time all trunks simultaneously in use exceeds threshold?	Select **Yes** to raise an event if the percentage of time that all trunks are simultaneously in use exceeds the threshold you set. The default is Yes.
Threshold - Maximum percent of time all trunks simultaneously in use	Specify the highest percentage of time that all trunks can be simultaneously in use before an event is raised. The default is 1%.
Event severity when percent of time all trunks simultaneously in use exceeds threshold	Set the event severity level, from 1 to 40, to indicate the importance of an event in which the percentage of time that all trunks are simultaneously in use exceeds the threshold. The default is 15.
Data Collection	
Collect data for percent of time all trunks simultaneously in use?	Select **Yes** to collect data for charts and reports. If enabled, data collection returns the percentage of time that all trunks were simultaneously in use during the monitoring period. The default is Yes.

Recommended Knowledge Script Group

The following Knowledge Scripts are members of the AvayaCM recommended Knowledge Script Group (KSG).

- CallActivity
- CPU_Usage
- ESS_Status
- H248GatewayStatus
- LSP_Status
- RegisteredResources
- SecurityViolations
- SystemUptime

The parameters of all scripts in the KSG are set to recommended values. To run all of the recommended scripts at one time, click the RECOMMENDED tab and run the AvayaCM group on a Communication Manager resource.

Run the KSG on only one cluster at a time. Running the KSG on multiple clusters all at once hinders the proxy agent's ability to spread out processing over time. You can monitor multiple clusters by running the KSG on the first cluster, and then repeating the process for each additional cluster.

The AvayaCM KSG provides a "best practices" usage of AppManager for monitoring your Communication Manager environment. You can use this KSG with AppManager monitoring policies. A monitoring policy, which enables you to efficiently and consistently monitor all the resources in your

environment, uses a set of pre-configured Knowledge Scripts to automatically monitor resources as they appear in the TreeView. For more information, see “About Policy-Based Monitoring” in the AppManager Help.

A KSG is composed of a subset of a module’s Knowledge Scripts. The script that belongs to a KSG is a different copy of the original script you access from the AvayaCM tab. If you modify a script that belongs to a KSG, the parameter settings of the original script in the AvayaCM tab are not affected.

When deployed as part of a KSG, a script’s default script parameter settings may differ from when the script is deployed alone. The default settings of a script within a group depend on its monitoring purpose within the larger group, and on the intended monitoring scope of that group.

If you modify or remove a script associated with the AvayaCM KSG and want to restore it to its original form, you can reinstall the AppManager for Avaya Communication Manager module on the repository computer or check in the appropriate script from the `AppManager\qdb\kp\AvayaCM\RECOMMENDED_AvayaCM` directory.